Modelling Defence: A Challenge to System Dynamics

Mike Young and Robert Bailey
CORDA Limited
Biwater House
Portsmouth Road
Esher, Surrey
KT10 9SJ, UK

Abstract

CORDA has developed SD Models for the UK Ministry of Defence over several years. This paper surveys a number of defence applications where Stella and iThink have been used, either directly to predict results for a study, to examine a concept, or to create a prototype which can be used to determine a functional specification.

Two of the models are described in more detail to illustrate how we overcame limitations either in the SD paradigm or in the current programs. The paper summarises a number of "lessons learnt" which it is hoped will prove of relevance to other practitioners and which may influence future program developments.

Modelling Defence: A Challenge to System Dynamics.

INTRODUCTION

CORDA (Centre for Operational Research and Defence Analysis) is one of Britain's largest specialist Operational Research consultancies. Over the last few years System Dynamics (SD) has featured in much of our work and we have developed a considerable respect for its power. This paper aims to present insights gained by CORDA through several years experience of working with SD in the modelling of defence problems.

Our work has concentrated on studies using the continuous flow packages, STELLA and iThink. Our paper concentrates on the problems encountered in the practical development of models using this software, and assumes at least a passing knowledge of the packages.

We believe that the advantages of these packages lie in:

- their speed and flexibility as model building tools in their own right
- their usefulness in the formative stages of a project, in clarifying the requirement and in building up a specification. This is true even if the model is eventually built using a different method

The paper is divided into three sections. The first reviews a number of models we have built, or considered building, using STELLA or iThink. It then looks in more detail at two models, which are used to examine the strengths and limitations of using the programs, and to illustrate how some of the modelling constraints were overcome. In the final section, we describe some of the lessons learned from our experience and draw conclusions about the utility of this modelling approach. Our views are those of practitioners required to produce quantitative outputs from the models built.

EXAMPLES OF MODEL BUILDING

In Table 1 we list examples of models that CORDA has produced over the past 5 years in which STELLA or iThink has either been used or considered. We briefly characterise each model application and list the advantages and disadvantages of approaching each via STELLA or iThink.

Models in the first group were implemented in STELLA, iThink's predecessor. The second set of models were built using iThink, and took full advantage of the enhancements of the new package. These models are larger and more detailed than the first set.

The final set of models are those for which SD was considered for model implementation, but not used. However, SD methods were used in the formative stages of the modelling to identify the influences that should be modelled and thus help specify the final model.

Problems Encountered

All the models produced using the SD approach were successful at the level of modelling to which they were taken. As a company we have found that our clients are primarily interested in using the technique to produce "hard" numerical outputs rather than the "soft" qualitative outputs with which the SD methodology is more naturally associated. There were bound to be areas therefore where implementation in STELLA and iThink presented us with challenges. These are outlined in the "disadvantages" column in Table 1. The biggest problems encountered were:-

(1) **Indistinguishability of the entities**. This is, of course, a basic characteristic of continuous flow simulation modelling. However it may be desirable, as would be the case in discrete simulation modelling, to give entities flowing around the system "labels" with specifying attributes such as whether or not an entity has been through a particular process.

(2) Difficulty in representing decision making - particularly:

- decisions dependent on the past or the forecast state of entities rather than the current state. There is no entity linking or forecast entity states to refer to
- decisions which depend on distinguishing the different entities. For example it is not possible to select "the oldest" or "the youngest" from a batch

Table 1 Models Considered for SD Implementation

Application area	Characteristics	Advantages of Stella / Ithink	Disadvantages of Stella / iThink	Final implementation
Tank Engagement Model	Models a series of tank engagements. Survivors from one engagement go on to fight in the next. Purpose was to evaluate tank options.	Simple way of modelling mutual attrition. Quick to build. Good interface.	Separate sub modules had to be built for each engagement. Restricted to one overall scenario. Realistic modelling of decisions on when to fight was not possible.	Stella
Artillery Model	Models the direct fire battle, deployment, engagement and supply of artillery and ammunition, in terms of a continuous (rather than discrete) simulation.	Enables representation of feedback between eg ammunition expenditure and rate of advance of enemy forces.	Difficulty of modelling geography. Need to repeat code for each major ammunition type.	Stella
Model of Torpedo In-service Support	Flow of torpedoes through maintenance facilities. Continuous model of a discrete system.	Ideal for investigating performance characteristics of system as a function of maintenance policy and level of facilities	Difficult to model resource allocation algorithms with elegance	Stella
Air Defence of United Kingdom Model (Version 1)	Models raids by enemy bombers and escorts, engaged by outer and inner air defence layers. Continuous simulation model. The interaction between many air defence assets e.g. radars, fighters, SAMs, airfields modelled	Enables modelling of feedback between enemy attacks on infrastructure and airfields, their recovery and repair and subsequent defence effectiveness	Difficulty of representing events on two different timescales - minutes *during* raids and days *between* raids. Difficulty of modelling geography. Need to repeat code for each class of target.	iThink 1
Air Defence of United Kingdom Model (Revised version)	As before with enhanced representation of geography and of air defence assets. Timestep changed to correspond to a raid.	Allows simple modelling of tactical decisions - eg proportion of enemy raids targeted against air defence infrastructure vs. strategic targets; proportion of defensive fighters on Combat Air Patrol vs. ground alert.	Difficulties in representing alternative scenarios and in changing decision rule data.	iThink 2.2.1
Model of Generation and Deployment of Mine Countermeasures Vessels (MCMVs)	Models flow of two different classes of MCMVs through refit and repair cycles, followed by deployment to two different theatres.	Permitted fast building and test of models representing five different scenarios, corresponding to different levels of commitment. Enabled representation of essential feedback loops.	Need to replicate code for each class of MCMV. Difficulty in attaching unique attributes to each entity (e.g. time since last refit). Difficulty in representing decision logic, and in changing embedded data.	iThink 2.2.1
Model of Generation of Naval Forces	Flow of different ship types through refit and repair, storing, trials and work-up at sea. Parallel modelling of ships' personnel through appointment, shore-based training and sea training	Flows could easily be modelled. Flow diagrams using iThink were useful in the conceptual stages of the model, even though the model was not implemented in iThink.	Need to represent unique attributes for each ship. Model required extensive resource allocation and data handling facilities. Model was required to be predictive.	Excel for prototype.
Model of Generation of Air Forces	Similar flows to above based around aircraft fleets, personnel and flying and combat training.	Similar to the above, but the resource allocation is simpler, and the feedback loops between training and aircraft preparation stronger, than in the naval model. There is no need to attach unique attributes to each aircraft, only to different aircraft *fleets*.	Resource allocation rules still difficult to represent in SD tools, and data handling facilities are not adequate to meet the needs of a predictive model.	Proposed to develop in an object oriented simulation shell.
Amphibious operations off load model	Model of the unloading of landing ships during amphibious operation. Represents movements of landing craft and helicopters from ship to shore	Suitable for modelling of flow of personnel, material and stores.	Difficult to represent the essence of the off-load problem - allocating the right stores to the right landing craft in the right order.	Object orientated simulation shell.

P(3) Representation of time. Normally this is a natural advantage of SD packages. However if a period when events are widely spaced in time is followed by one where they are concentrated into fast moving encounters, choice of an appropriate time step becomes a problem.

(4) Representation of detailed geography. The ability to represent geography is important in many defence applications. SD packages are not in general designed to do this, though it can be achieved, at a cost in terms of complexity, by replicating code to represent events in different areas.

(5) Difficulty of modifying data. It is tedious to modify large amounts of embedded data, or to undertake a series of parametric runs of a model and save the results. Although iThink has the "Sensitivity run" options, none the less we find the procedure quite cumbersome to automate.

Problems (1), and to an extent (2) and (3) above are a result of adopting a continuous flow rather than a discrete simulation package. However if suitable work-arounds to the problems can be found, we can still achieve very cost effective modelling within the SD paradigm. These work-arounds are examined further in the next section.

A CLOSER LOOK AT TWO OF THE MODELS

Two models are selected and described in more detail, with the aim of illustrating the problems encountered above and discussing the form of our work-around solutions.

Air Defence of United Kingdom Model

This model was developed as part of the MOD's Measurement of Defence Output programme, which aims to measure the output of defence programmes in terms of the capability to perform a number of Military Tasks. The aim of this model was to estimate the RAF's ability to defend UK airspace against a series of attacks which, when the model was developed, were based on a threat from the Warsaw Pact. The measure of output used was success in preventing damage to "strategic targets" such as military installations and ports.

The first step in model specification was to draw up an influence diagram showing the basic feedback loops relating the attackers and defenders. Figure 1 is a simplified version of this diagram.

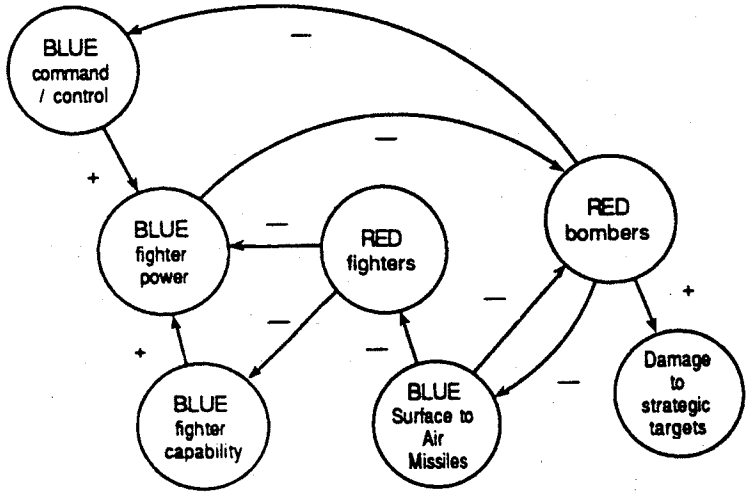

Figure 1 – Air defence of the UK – basic interactions (BLUE and RED denote own forces and enemy forces respectively)

The model was then expanded to take a more detailed look at the combat capability of the defending fighters which was dependent on the quality of the aircraft, the skills and training of their crews and the effectiveness with which they were allocated to tasks. These various factors had to be modelled explicitly for their effect on the overall output measure to be assessed. Figure 2 gives a simplified view of how they were incorporated in the model.

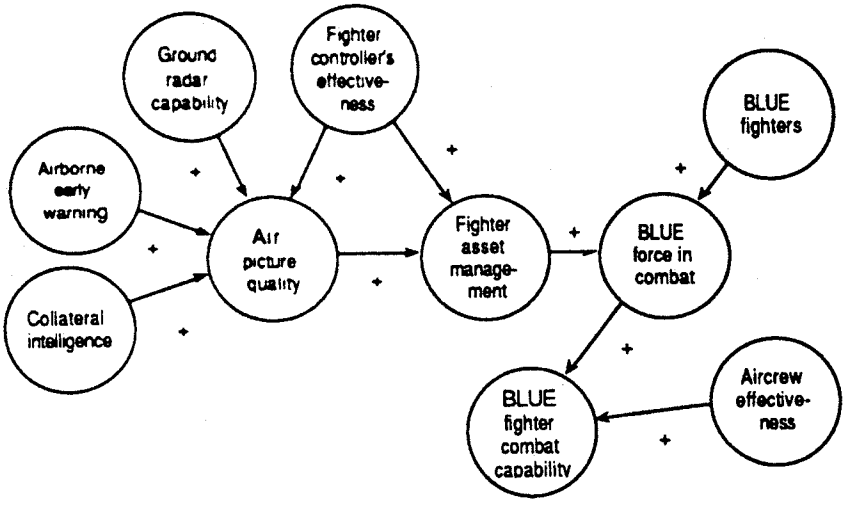

Figure 2 – Expansion of influences affecting BLUE combat capability

By developing influence diagrams in this way, we were able to agree with the sponsors of the work what factors were needed, and how they should be brought into the model. This ability to communicate the proposed logic of the model is a particularly attractive feature of the SD paradigm.

To meet the needs of the sponsors, three key features needed to be addressed. These were:

1) **Representation of decision-making.** Both sides would face important tactical decisions about the way they conduct the battle. For RED, the ultimate aim would be to inflict damage on strategic targets. However, their most effective tactic might be to attack the BLUE air defence infrastructure, and hence later to be able to bomb strategic targets with much less opposition and fewer losses. For BLUE, there were a number of decisions to be made about the management of its fighter force. Table 2 lists some of the decisions which needed to be modelled on both sides.

Table 2 Tactical Decisions to be Modelled

RED
Do I attack the "strategic targets", the Air defence infrastructure or the Surface to Air Missiles?
How long should I wait between raids?
Is there a particular area of the UK which it is best to raid this time?
Do I launch low level raids (to avoid ground radars, but with a smaller payload) or high level raids (easier to spot with ground radars)?
BLUE
In which area of the UK should I concentrate my fighters?
Do my fighters attack the bombers or their escorting fighters?
What percentage of fighter aircraft should be kept on the ground in a state of high alert and what percentage should form standing Combat Air Patrols?

These decision processes were represented in the model by simple allocation rules triggered by specific data combinations. The software packages used did not allow for easy inspection or modification of these rules.

2) **Representation of time.** Air raids happen intermittently. There are periods of recuperation and repair which may last several days, punctuated by periods of intense activity which may last only a few hours. This means there is a problem in choosing the time step: a small time step would lead to an unacceptably long run time, and a long time step to insufficient representation of detail during a raid.

The solution to this problem was to model in units of "air raids", rather than in hours, days or weeks. Each time step of the model represented one raid and the subsequent recuperation period. All the activities during a raid (e.g. flow of aircraft from one air defence zone to another) were calculated using converters, so the results of each raid took only one time step to calculate. In contrast, the repair and recuperation activities which take place between raids were modelled using the standard iThink procedures, except that the rates of flow of these activities had to be multiplied by the time between raids to determine the amount of activity in each time step. Using this procedure, it did not matter if the air raids were unevenly spaced. The only assumption made was that no recuperation activities took place during the course of a raid. A simplified representation of the flows of RED bombers resulting from this approach is shown below.

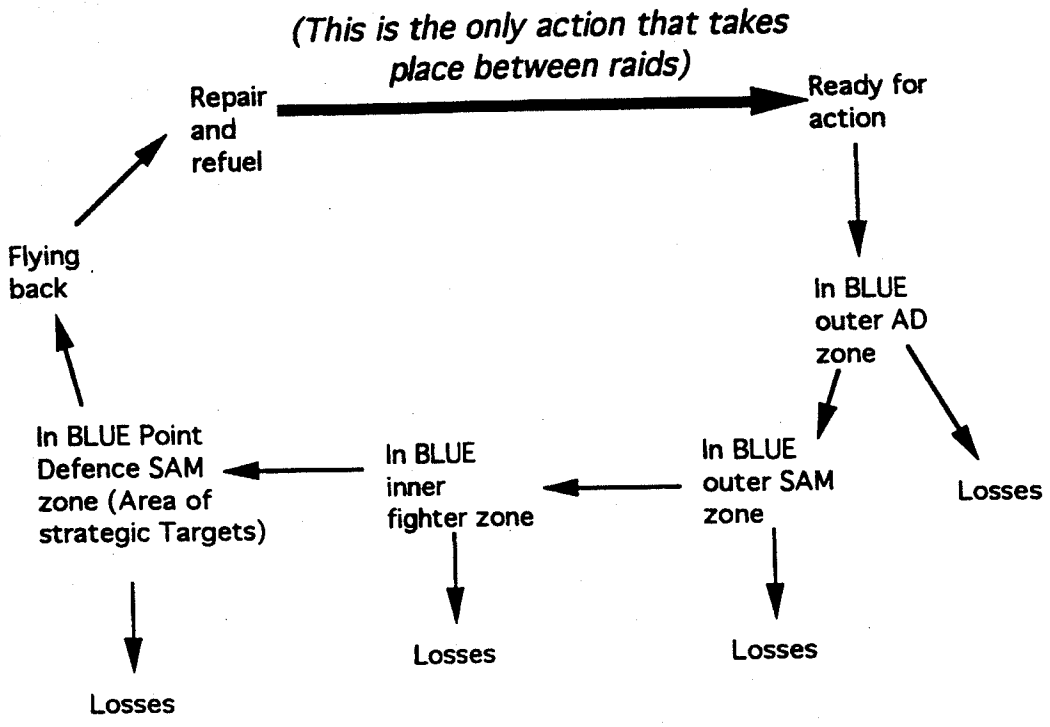

Figure 3 - Flows of RED Bombers

Figure 4 shows in outline the way in which time was represented in the model.

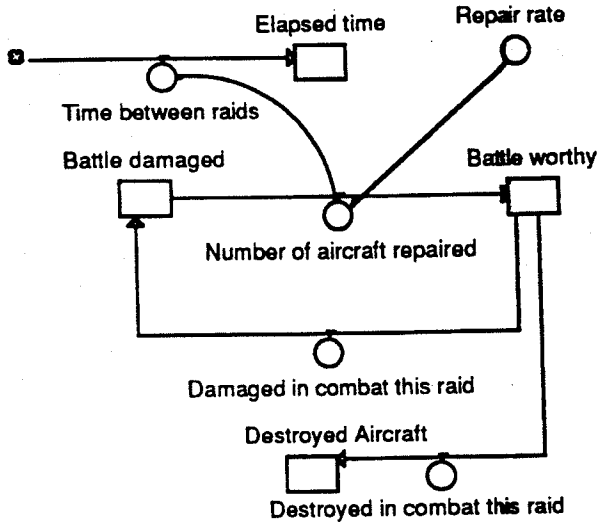

Figure 4 - Time Representation

3) **Representation of geography.** Version 1 of the model represented the UK as a single air defence region, effectively modelled as a single point. However, it was considered that this might overestimate of the capability of the defence, since both targets and fighter bases are distributed across the UK, and so a fighter based in one area may not be able to intercept bombers attacking a target in a different area.

The revised version of the model attempted to increase model fidelity by representing the UK as three regions and directing each air raid against one of these regions. The resulting changes to the modelling assumptions are outlined in Table 3.

Table 3 Assumptions about Geography

Original Model	Revised Model
A RED raid could attack any part of the UK.	A RED raid could be concentrated against one third of the UK.
Every BLUE fighter could defend against every raid.	BLUE fighters in the "North" could not defend raids against the "South" and vice-versa. BLUE fighters in the "Centre" could help against any raid, but with penalties.
	After each raid BLUE assets were redistributed evenly across the three regions.
Strategic targets always existed. Their supply was never exhausted, they could never be repaired.	Strategic targets always existed *in all three regions*. Their supply was never exhausted. They could never be repaired. .

Tripling the number of regions more than tripled the complexity of parts of the model. In particular, the decision making was made much more complex, as there were many more tactical options available to both sides. Better fidelity is not necessarily gained by increasing detail! – in this case a lot of work was required to validate the expanded representation.

Development approach. The model was built as two sequential prototypes. A specification in terms of influence diagrams and entity flows was drawn up at an early stage, together with examples, in graphical form, of the nature of output the model was expected to produce. The model was tested and modified to broadly reflect those expected behaviours.

Influence diagrams were also useful in explaining the model to sponsors. Influence diagrams provided a qualitative specification of the influences but could not say how important these influences were. An initial prototype model enabled the sizes of these influences and the model behaviour to be established early on, using estimated data.

Mine Countermeasures Vessels (MCM) Activity Cycle Model

This model was developed to support a study to examine the degree to which future options for the Royal Navy's Mine Countermeasures (MCM) force would be able to be meet the range of demands that may be placed on that force. The model was built to assess the ability of a given force to meet a requirement specified in terms of numbers of vessels to be made available for operations in different locations.

The model takes into account the fact that the Mine Countermeasures Vessels (MCMVs) are not always available for military operations, since they must regularly undergo various maintenance activities. It also allows for commitments to non-warfare tasks such as fishery protection.

The model was conceived in terms of entity flows and queues. It is based on a circuit representing the particular activities which the MCMVs are performing at any given time, duplicated so as to allow two classes of MCM vessel (the "Hunt" and "Sandown" class) to be distinguished from one another. A basic circuit for a single MCMV type and a single deployment is shown in Figure 5 below:

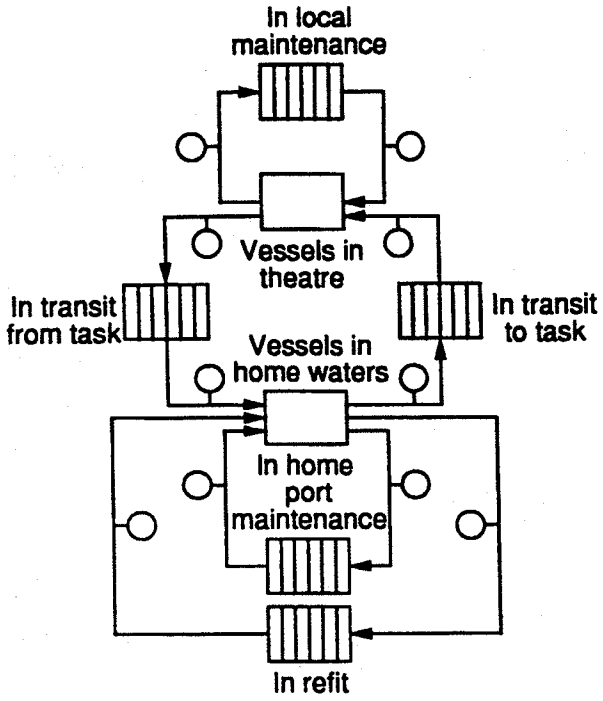

Figure 5 Circuit of Entity Flows and Queues

This basic activity cycle model was then expanded considerably to take into account:

- simultaneous commitments in several geographical areas
- ongoing peacetime commitments, such as fishery protection
- two types of maintenance period, of different lengths (Docking and Essential Defects (DED) and Refit)
- workup periods before deployment after completing refit

The times for these activities were different for the different ship classes.

This model differs from the Air Defence of the UK model in that it was naturally expressed by entity flows rather than by influence diagrams. There were fewer influences and more "queuing", and the model could have been built using another paradigm. iThink was chosen as the modelling tool because of its speed of development, on the assumption that representational detail at the single ship level would not be needed.

A number of interesting problems needed to be addressed in implementing this model:

(1) **Distinguishing and scheduling entities.** The indistinguishability of the entities was a major problem: the total number of ships in a scenario was fairly small, and it would have been useful to be able to represent them individually and give them attributes such as "time since last in refit". This in turn would have made their scheduling easier to model.

(2) **Embedded data.** In iThink the data used is embedded in the model. This makes it tedious to keep track of the changes from run to run. A useful package enhancement would be the ability to read or write data from spreadsheets.

(3) **Formulating the decision logic.** A crucial aspect of the model was the logic that allowed ships to be assigned to different tasks. This depended on circumstances and varied with time. For example if a simulated "crisis" arose and ships had to be sent overseas, the model would need to send

ships of the required types in the required numbers, along with the necessary reliefs. If there were not enough ships, the model would have to decide how it was going to compromise in meeting the demands.

These decisions were difficult to simulate in iThink. This was because decisions would ideally be based on the past history of the ships, for example how long a ship has been in a particular activity. Although conveyors can be used to keep track of how long a ship had been in a particular activity, it was difficult to model the fact that ships sometimes had to abandon an activity prematurely, in order to perform a higher priority task. This complicated the decision as to when a ship would need to stand down for its next maintenance period. We managed to work around this problem, but it required a great deal of fine tuning to obtain satisfactory model behaviour.

In the event the SD paradigm was a fairly natural one. The model was successful in that it served the study needs. The problems arose when we wished to make decisions based on individual ship histories rather than treating ships collectively.

LESSONS LEARNED

In our discussion of the two rather different models used as illustrations in the previous section, we identified five technical areas which caused difficulties in model building using these packages:

a) distinguishing entities in the model.

b) representing decision-making.

c) representing time when activities are intermittent. This can be solved in this case by measuring time in units of activities (in our example air raids).

d) representing detailed aspects of geography.

e) modifying data embedded in the model.

Items a) and d) probably represent inherent limitations of the SD approach. If greater modelling fidelity in these areas is needed, then a different simulation environment should be used.

Item c) was successfully addressed.

Items b) and e) derived from the then current limitations of iThink, and could be overcome by improvements to the package.

DEVELOPMENT ISSUES

Our experiences raises a number of model development issues. Perhaps the key issue relates to the production of a functional specification. Normal software life cycle disciplines require that this, together with a design specification, should precede model implementation, thus ensuring a clear focus and a controlled development. Whenever possible our models are built in this way. However, often the initial understanding of the problem being addressed is not sufficient to allow a functional specification to be drawn up without some investigation. It is in these circumstances that we have found SD modelling especially valuable. Prototyping in SD helps to clarify the functional specification and to build up an understanding of the response properties to be expected of the model. Whether the final model is implemented using an SD or another paradigm is a decision that can be taken with the benefit of a deeper understanding of the required model's key characteristics.

CONCLUSIONS

The SD paradigm and the packages we have used are powerful, but limited tools for the modelling of defence applications. Their strengths - the power and clarity with which relationships can be expressed and communicated to users, and the speed with which models can be developed - make them attractive for the investigation of new applications. A number of work-arounds we have adopted increase the scope of application of the programs.

It is hoped this paper proves to be of relevance to other practitioners and serves to influence future SD package developments.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the considerable assistance received from colleagues in CORDA in reviewing and improving this paper, and from DOAC and DSc(L), MOD, for their permission to illustrate from models developed for them under contract.