Optimising Workforce Structure
the System Dynamics of Employment Planning

Keith T Linard
Senior Lecturer, University College, University of New South Wales
Australian Defence Force Academy CANBERRA ACT 2601 AUSTRALIA
Tel: -61-2-6268-8347 Fax: -61-2-6268-8337
k-linard@adfa.edu.au

Captain Mark Blake
Australian Regular Army
Mark.blake@deo.defence.gov.au

David Paterson,
Computer Science Corporation (Australia) Pty Ltd
dpaters3@csc.com

This paper illustrates key features of an enterprise employment simulation which integrates a system dynamic feedback model with a cost-effectiveness optimisation capability utilising genetic algorithms. Its core is a 3-dimensional array structure tracking staff numbers by rank, by time-in-rank, by years-of-service.

In May 1997 an internal review of Australian Army employment recommended implementation of more flexible employment practices similar to contract based private sector approaches. Because of the history of instability due to 'boom-bust' recruiting over the previous decade, and because the changes were a dramatic departure from current practice, the Defence Department contracted for the development of a system dynamics model to test their consequences.

The resultant model, which could readily be adapted to non-Defence use, can identify, given user specification of any mix of employment rules, the likely patterns of employment behaviour including: resultant time-in-rank and years-of-service profiles; ability of a Unit to fill all positions to target strength; ability to fill promotional positions within normal rules for substantive promotion; need to fill promotional positions using rules for temporary promotion or transfer from outside; necessary recruitment pattern to sustain target strength.

Background to the Army Employment Model Project

From time to time changes are proposed to the methods of entry into and conditions of service in the Armed Services. Whilst the impact of minor changes can be reasonably predicted by personnel experts, significant changes can result in longer term unexpected and undesirable outcomes. This is especially so in complex organisations such as the Armed Services which involve complex interactions over time between the organisational elements.

Thus a rapid turnover employment policy may result in an organisation structure in 10 years time which has a large emergency reserve, but which cannot supply trained

NCO's to handle rapid mobilisation. Conversely, a low turnover employment scenario may diminish promotion opportunities which in turn impacts on morale. Policy switches between the extremes may produce totally unexpected consequences. Also, as the peace-time Army is fundamentally a training organisation, 'boom-bust' recruiting patterns result in successive peaks and troughs coursing through the training system for years after the initial event, causing considerable inefficiencies.

The specific catalyst for this project was the policy decision to move the Australian Army from a 'lifetime career service' to a workforce with more flexible rules for entry and exit. Given the 'boom-bust' consequences of army recruiting policy over the previous decade there was concern to understand the implications of alternative employment scenarios. The project contract specifically requested a 'system-dynamics model' and also specified that this model was to permit identification of the 'optimum' employment strategy.

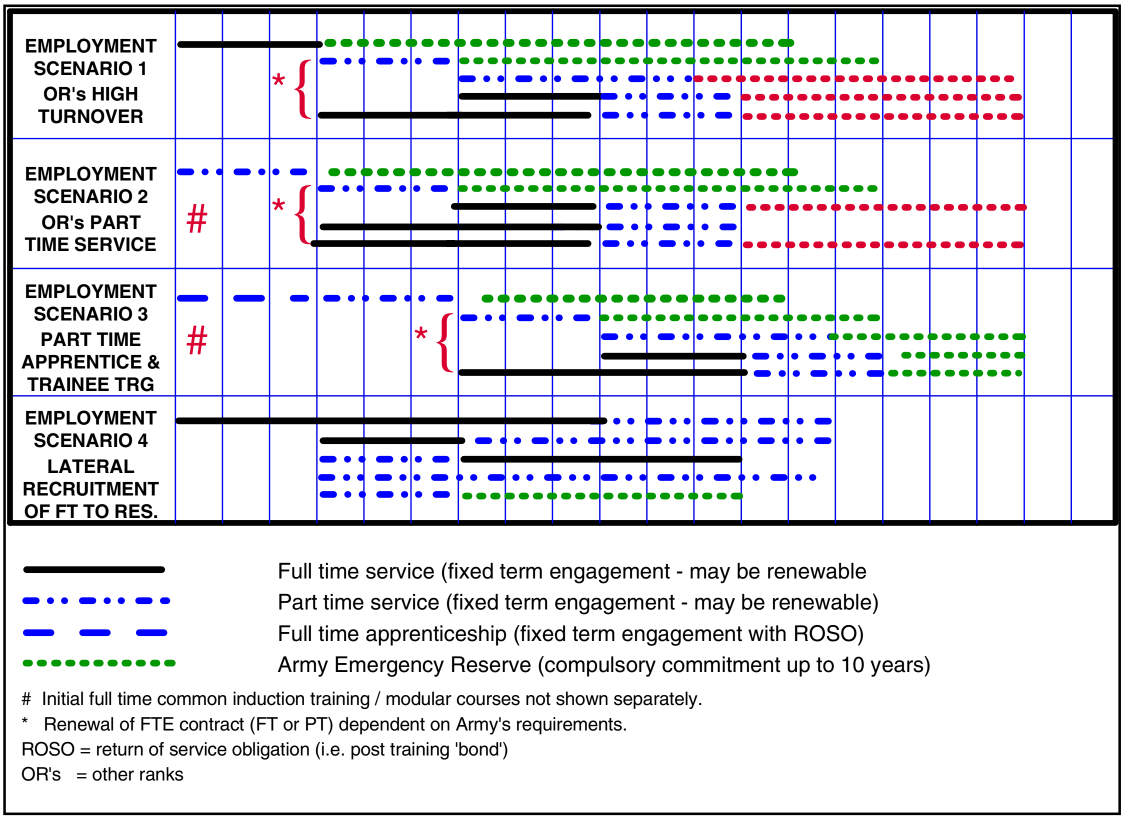


Figure 1: Employment options over 20 year time horizon

The simulation tool was to allow personnel policy analysts to model diverse combinations of employment scenarios, testing them against policy criteria such as:

- ability to fill rank structure
- time in rank and years of service profiles
- cost structure
- outcome efficiency.

Having determined an appropriate combination of full and part time employment scenarios which would meet strength and preparedness criteria for the various employment categories, it would then become possible to address

The Nature of the Model

The simulation system comprised the system dynamics model, built in Powersim, and an EXCEL spreadsheet where the data for the different employment scenarios was entered. The Powersim model included:

- Full Time employment module
- Part Time employment module
- Army Emergency Reserve module
- Productivity performance measurement module
- Promotion and transfer rules module

Characteristics of the employment modules

The core of the model is the stock 'Workforce', which is a three dimensional array in which we maintain key attributes of personnel in each cohort. These attributes are rank, length of service and time in rank. Personnel are recruited only at the lowest level, and 'spiral up' the array incrementing each additional year of service and time in rank and each promotion through a possible six ranks.

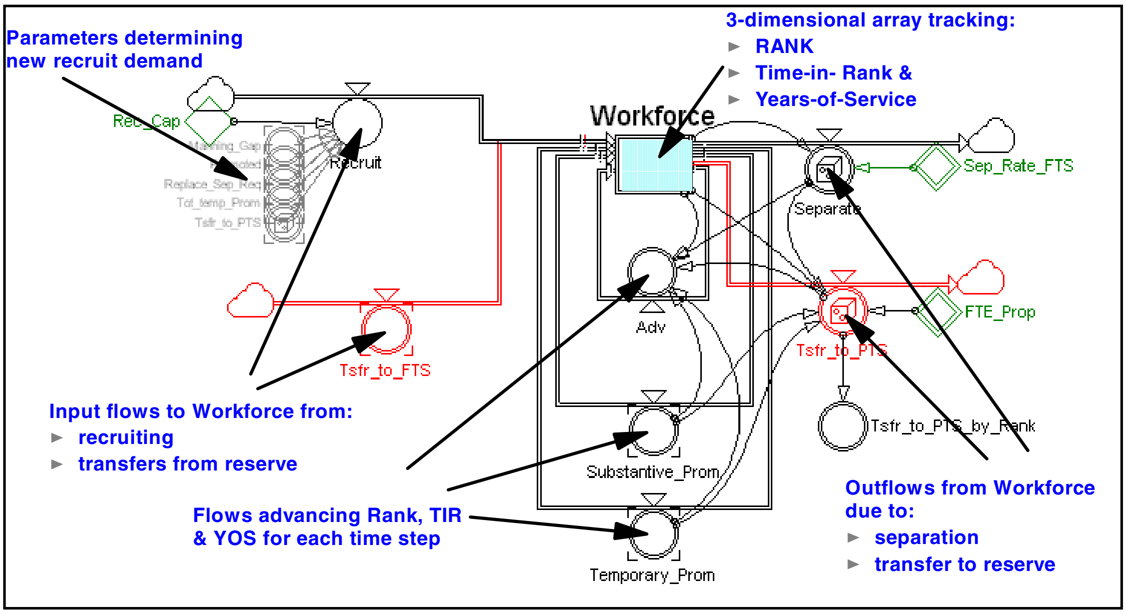


Figure 2: Full time employment module

The use of Powersim's powerful array structures results in a stock-flow diagram which is sufficiently 'simple' and uncluttered to use as a basis for validating the broad business rules with subject area experts and to use with senior managers in explaining counter-intuitive consequences of specific scenarios. At the same time it permits the capturing of critical organisational data. The 'simple' module in Figure 2 contains some 12,000 elements!

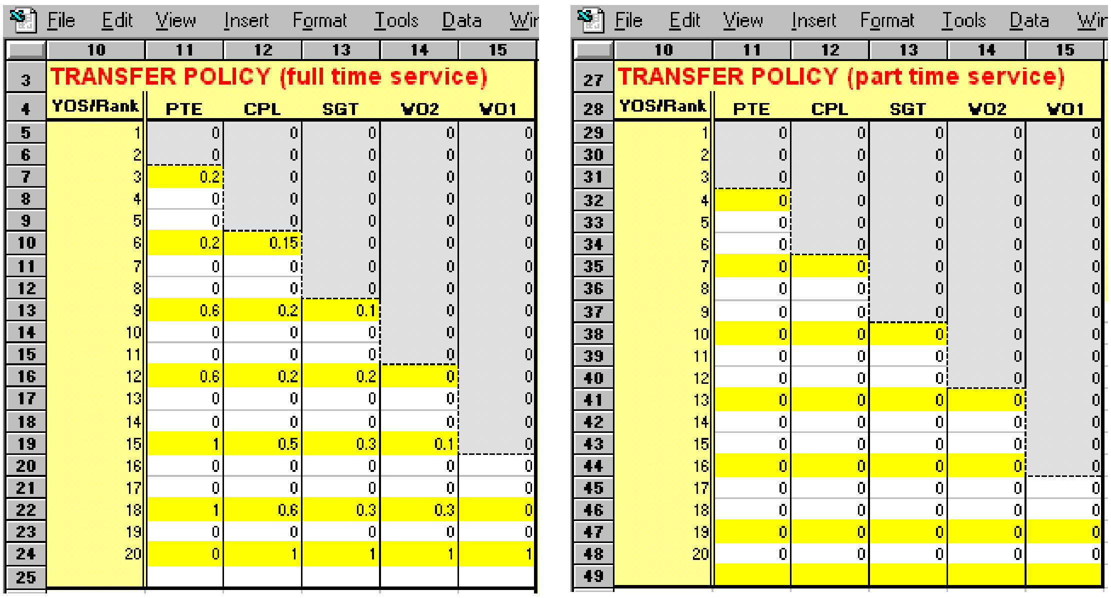
Scenario Building - Base Data and Employment Options

The decision was made to use spreadsheets for basic data entry and scenario specification because of the complexity of entering data directly into the array structures and also because the client's staff are very familiar with spreadsheets.

The first step in building an employment scenario is the specification of base characteristics of the Army unit or other aggregation. The following personnel policy parameters data items are required:

- Strength targets by rank for full time and, where appropriate, part time service.
- The promotion cohorts for each rank (i.e. minimum time-in-rank for substantive promotion).
- The proportion of each cohort likely to be considered suitable for substantive promotion.
- The proportion of each cohort likely to be considered suitable for accelerated promotion.
- Separation rates by rank, by years-of-service (held constant over time in order to focus on the effect of different employment scenarios).

The different employment scenarios depicted in Figure 1 (and indeed any variant thereof) are specified through the 'Transfer Policy' data entry blocks for full time and part time service illustrated in Figure 3.



TRANSFER POLICY (full time service)

YOS/Rank	PTE	CPL	SGT	VO2	VO1
1	0	0	0	0	0
2	0	0	0	0	0
3	0.2	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0.2	0.15	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0.6	0.2	0.1	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0.6	0.2	0.2	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	1	0.5	0.3	0.1	0
16	0	0	0	0	0
17	0	0	0	0	0
18	1	0.6	0.3	0.3	0
19	0	0	0	0	0
20	0	1	1	1	1

TRANSFER POLICY (part time service)

YOS/Rank	PTE	CPL	SGT	VO2	VO1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0

Figure 3: Data entry for alternative employment scenarios

Each cell in Figure 3 represents the percentage of the total numbers in a particular TIR cohort in the chosen rank who will be 'transferred out' from FT to PT service or from PT service to the Army Emergency Reserve (AER). Thus, an entry (in the 'FT' Data Entry Block) of 0.2 for privates at the 6 year mark means that 20% of all privates who reach 6 years TIR will not be offered continuation of FT service, but will be compulsorily transferred out to PT service or the AER.

In order to build a particular employment scenario the user specifies, for each 3 year band (it is assumed that contracts are for 3 years with possibility of renewal, but any number of years may be used) the proportion of each rank to be transferred to PT service.

A <u>high turnover model</u> would be developed by increasing the rate at which staff at given ranks are transferred out. For example, all privates remaining at that rank after 9 years and all corporals remaining after 12 years could be transferred.

The three sets of graphs below are illustrative of the scenario information produced.

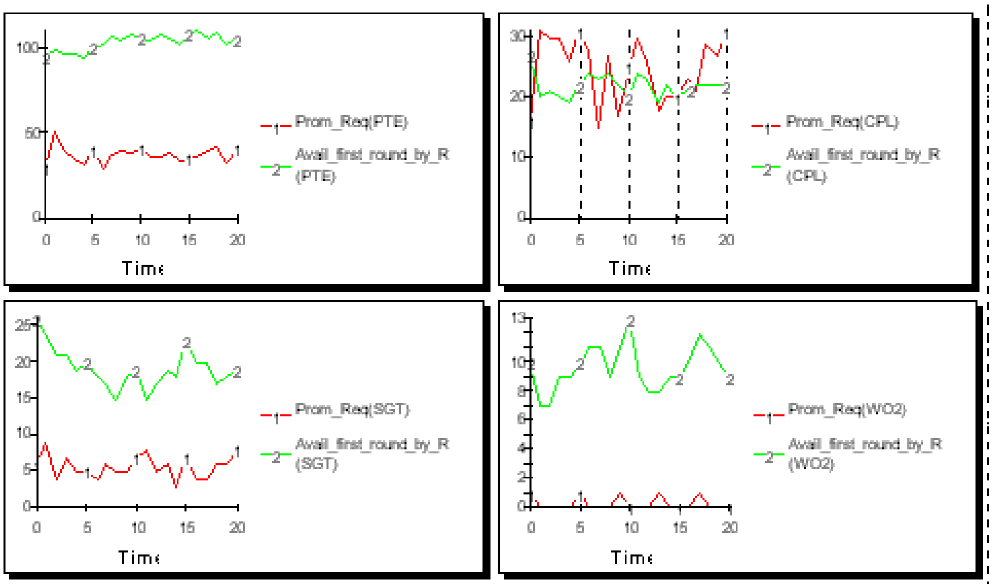


Figure 4: Detail of numbers available for promotion versus numbers required

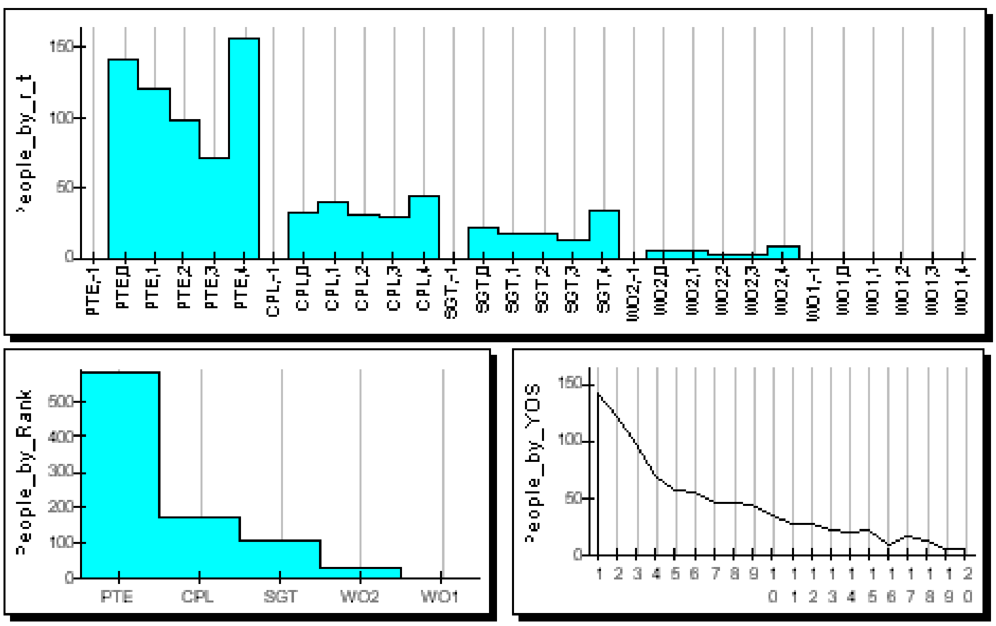


Figure 5: People by rank by TIR / People by YOS (most recent time step)

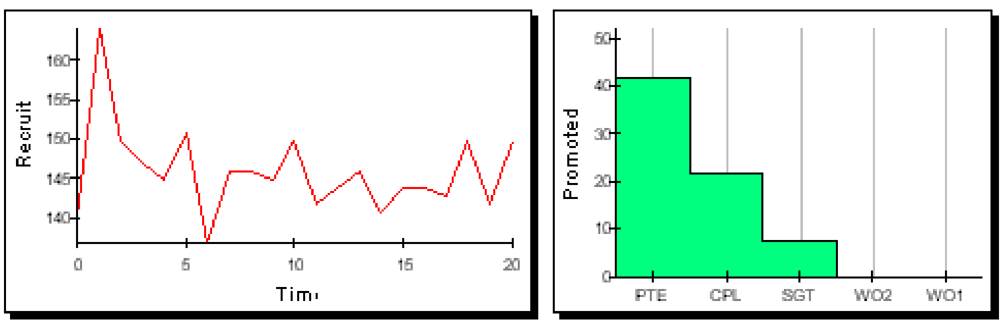


Figure 6: Recruitment and promotion patterns

Application of Optimisation to System Dynamics Modelling

'Traditionally, System Dynamics has relied on the use of intuition and experience by system owners and analysts to help design policies for improving system behaviour over time.'[1] First, modellers would decide on a reference mode which is representative of a system's behaviour in the real world. 'The dynamic behaviour (and hence accuracy) of the model is assessed in terms of its feedback mechanisms.'[2] If the model does not produce the reference mode behaviour then it's structure and parameter values are manually altered until it does. Wolstenholme [1990] summarises the traditional process of System Dynamics model design using the feedback loop shown at Figure 7.

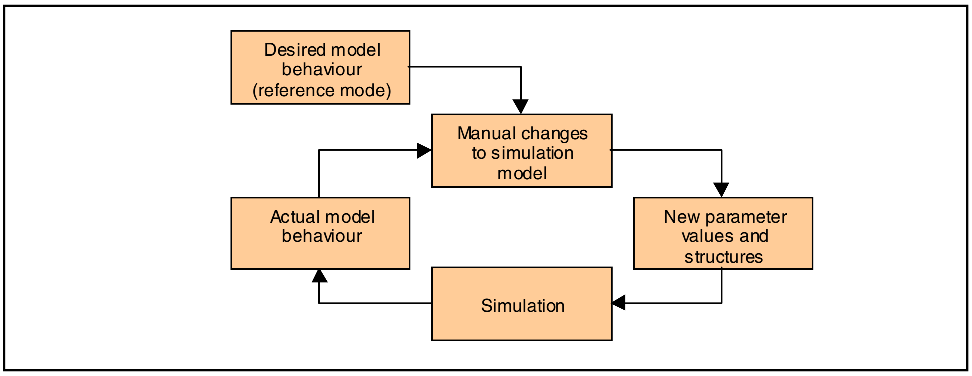


Figure 7: Tradition system dynamics modelling sequence

Changes made to a model's parameters or structure in an effort to seek the desired behaviour of the model are done on a "trial and error" basis. 'The drawback is, however, that there is always a nagging doubt that, had one tried only one more experiment, something even better may have been found.'[3] Sholtes [1994] also points out that policy makers often tend to avoid the use of System Dynamics models because of the expertise required to operate them and the time required to determine the necessary changes to the model to discover the 'best policies.'

At first glance, it would appear that automating the process of policy option selection could be readily achieved by developing a routine that ranges through all possible combinations of each. This is akin to enumerative search techniques, whereby the entire 'landscape' of possible solutions is tested for optimality. However, as Coyle [1996] suggests, there are potentially an infinite number of possible combinations and conceivable numerical values of parameters. Therefore, this approach is unrealistic.

A Few Words on Genetic Algorithms

Genetic algorithms were invented in the 1960's at the University of Michigan by John Holland. The genetic algorithm search procedure is based on the Darwinian principles of survival of the fittest. Optimisation is achieved through the emulation of biological evolution, and terms such as population, reproduction, genes, chromosomes, and mutation have been borrowed from this natural process to describe the genetic algorithm method.

Consider a domain such as that represented schematically in Figure 8. In approximate terms, the algorithm initially generates a population consisting of a predefined (user declared) number of randomly generated solutions (that is, points that lie on the surface shown in Figure 8).

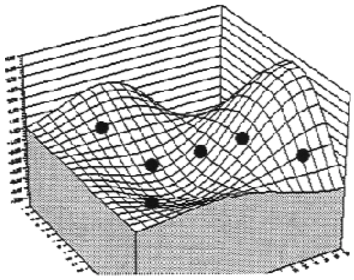


Figure 8: Domain of results - search for optimum

The 'altitudes' ('z' coordinate value) of each possible solution are compared, with those solutions that have higher altitudes (that is, 'fitter' solutions), retained for the recombination process. Recombination involves the random pairing of these retained solutions. Each pair then exchanges 'x' or 'y' coordinate values (known as the crossover process) to produce another two sets of coordinates. In this way, the original pair are considered to be parents and the newly generated coordinate sets are the children or offspring. The new generation formed by the children from each of the mating pairs is then assessed for fitness and manipulated in the same way that their parents were. This process continues for a fixed number of iterations or until a certain tolerance within the desired outcome is achieved (as defined by the user).

Sholtes [1994] demonstrated the application of a custom designed Genetic Algorithm routine to solving System Dynamics models by optmising the Kaibab Plateau Model. Sholtes made the prediction that advances in modelling software, combined with improved software inter-operability would eliminate the need to create custom routines. ‘In the future it should be possible to take your favourite model and import it into an optimisation program or have the genetic algorithms built directly into your simulation software.’[4] This indeed is the situation that has permitted the development of the optimisation of the employment model.

Irrespective of the type of optimisation routine that is applied to a System Dynamics model, the way that the two interact is essentially the same. Wolstenholme’s [1990] feedback loop for model design using optimisation shows how ‘optimisation in parameter space is achieved by interleaving simulation and optimisation.’[5] Keloharju explains that the method should not be seen as optimisation through repeated simulation, but simulation via repeated optimisation.

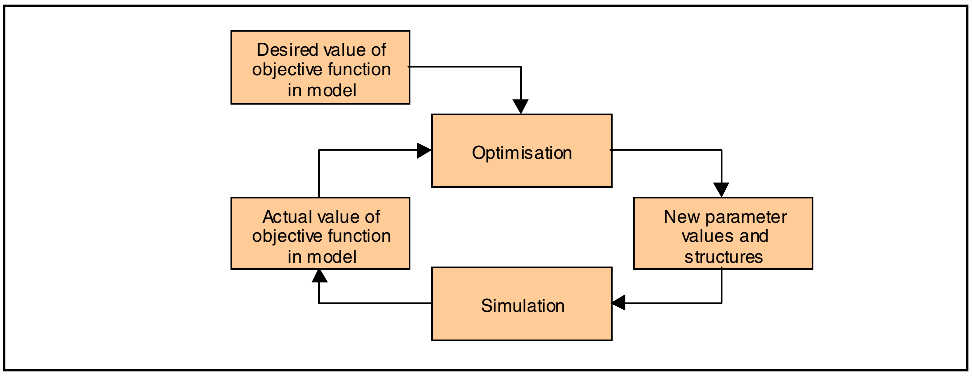


Figure 9: Model design using optimised System Dynamics

Optimisation Objectives for Modelling Employment Strategies

Notwithstanding the comments above, optimisation remains a controversial topic in the system dynamics community. We would argue that it is an invaluable tool in at least two restricted contexts.

First, we have found the optimisation process valuable in validation. We have found that the random process of selecting input variables occasionally produces results that should be impossible. Closer inspection finds a flaw in logic or in the specification of business rules. Manual testing had not picked up the problem because the input variables were not ones we would intuitively try.

More importantly, where there are potentially a very large number of decision levers, or a large number of possible 'positions' for those levers, optimisation can identify initialisation settings which are 'pretty good', from which 'what-if' analyses and sensitivity analyses can be undertaken.

Consideration of the 'Transfer Policy' options implicit in Figure 3 make it apparent that the user can specify any '% transfer policy' (in the range zero to 100%) for each year of service (from years 1 to 20), for each rank. In other words there are an *infinite number* of combinations and permutations of policy possible for each of the employment scenarios modelled.

Many of these combinations will yield a structure which cannot sustain the strength targets without cannibalising other units or which cannot achieve preparedness and mobilisation targets. Even if we discard all those combinations of transfer policy which lead to failures to meet boundary constraints, there will still be an inordinate number of 'feasible' solutions. 'Optimisation' allows us to identify an initial position for our policy levers which is 'pretty good' even if an 'optimum' as such does not really exist.

Optimisation presumes that there is an objective function to optimise

It may seem axiomatic that organisations know what they are trying to optimise, especially if they ask for the 'best' option. However our question to the client "What does your employment policy aim to optimise?" was met with a blank look. Army, in fact, could not specify optimisation criteria against which to judge the different employment scenarios. This, in our experience, in not uncommon.

In the absence of guidance from the client we included a 'placeholder' that could be replaced if and when Army identified an 'optimisation' employment objective. The simulation model has a crude employment cost-effectiveness measure based on output productivity and total salary cost. This productivity module consists of two key facets:

- productivity by years of service, and
- time on task (versus time on supervision and management etc)

Figure 10 illustrates these concepts. The productivity by years of service graph in illustrates a hypothetical trades employment category. An apprentice has minimal productivity for the 3 years in training; has a productivity in the first year after graduation of around 40% of that of the master tradesman; and thereafter gradually increases in productivity with years of experience.

The time on task graph shows that, at the rank of private, the majority of the working day is spent on task (allowing about a 20% overhead for routine military activities) regardless of YOS. Higher ranks, however, spend an increasing proportion of their time on supervisory and managerial duties. Combining the two graphs for the unit gives an average productive output potential. (There is, in fact, a dynamic relationship between productivity of subordinates and supervisory time spent by managers, but this was ignored in the first instance.)

The model can thus compare alternative employment scenarios, which may otherwise seem equally satisfactory, on the basis of their respective outcome efficiencies.

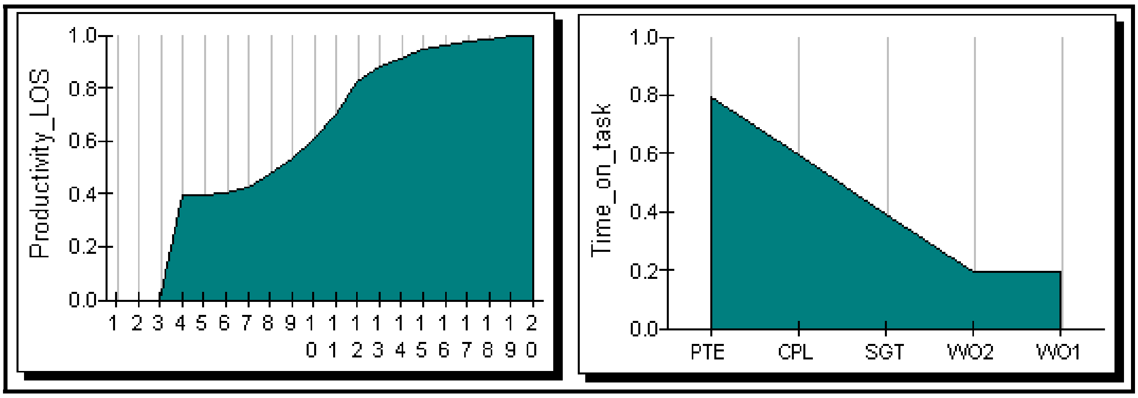


Figure 10: Productivity by Years of Service and Time-on-Task by Rank

The specification of productivity by years of service is essentially qualitative, based on the judgements of experienced NCO's and unit commanders. An ADFA research project is currently gathering estimates of this data for a variety of Army trades. The broad framework for time-on-task has been taken from the case presented to the Remuneration Tribunal in an Army pay claim. Again this needs to be validated on an trade by trade basis, as the characteristics can be expected to vary between the administrative, technical and arms trades.

The Optimisation Tools

The optimisation capability was achieved by integrating the genetic algorithm optimisation software 'Evolver' with Powersim and EXCEL. 'Evolver is a set of proprietary Genetic Algorithms which can be run as an add-in for Excel, although the Evolver solving methods can be used within many other applications.'[6] The model to be optimised is defined within an Excel spreadsheet. If the model is specified in another Windows-based application (such as Powersim), Excel acts as the medium of information exchange between that application and Evolver.

Broadly following Wolstenholme's [1990] framework (Figure 9) of the required interaction between an optimisation routine and system dynamics model, the Evolver, Excel and Powersim applications were combined in accordance with the following steps (see Figure 11):

- Evolver selects a population and updates the values of the model variables to be optimised. The model variables are defined in cells within Excel.

- These values are sent to Powersim and a simulation is conducted.

- Once the Powersim simulation is complete, the value of the objective function is returned to Excel to enable Evolver to assess it's fitness and generate offspring accordingly.

- Evolver selects new values for the variables to be optimised (the offspring), and amends the appropriate cells in Excel.

- If the stopping criteria specified to Evolver is not met, go to Step 2.

To facilitate the interaction between the software, the cells in Excel that contain the range variables and the objective function value must be linked by DDE to corresponding variables in Powersim.

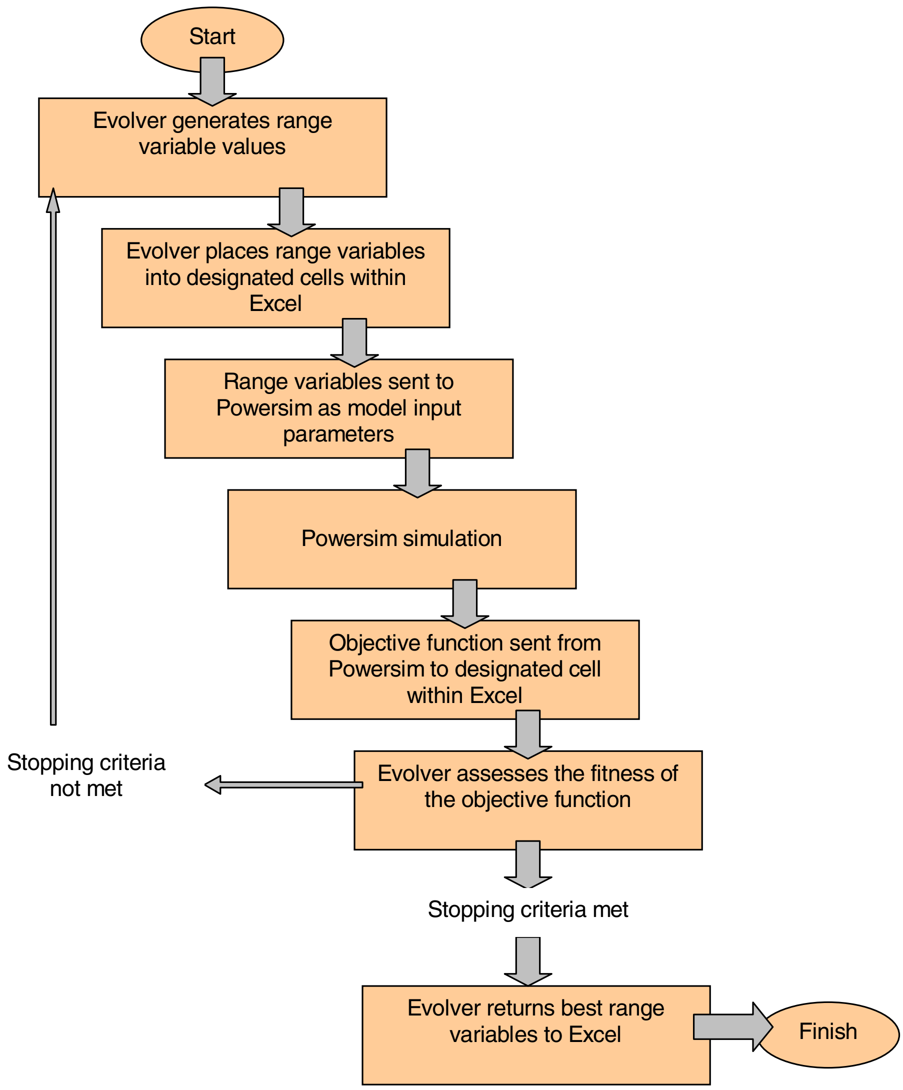


Figure 12: Basic optimisation strategy using Evolver, Excel and Powersim

The data flows between the packages are illustrated in Figure 13. The Powersim model had some 35,000 elements, and approximately 50 policy variables were being varied with each iteration. In addition, boundary condition tests were applied against some 20 factors. Each complete iteration took just under 1 minute on a 233MHz Pentium with 128 Mbyte of RAM. Typically the model reached stability within 2,000 to 2,500 iterations (1.2 to 2 days), although in the tests the model was typically run for 10,000 iterations.

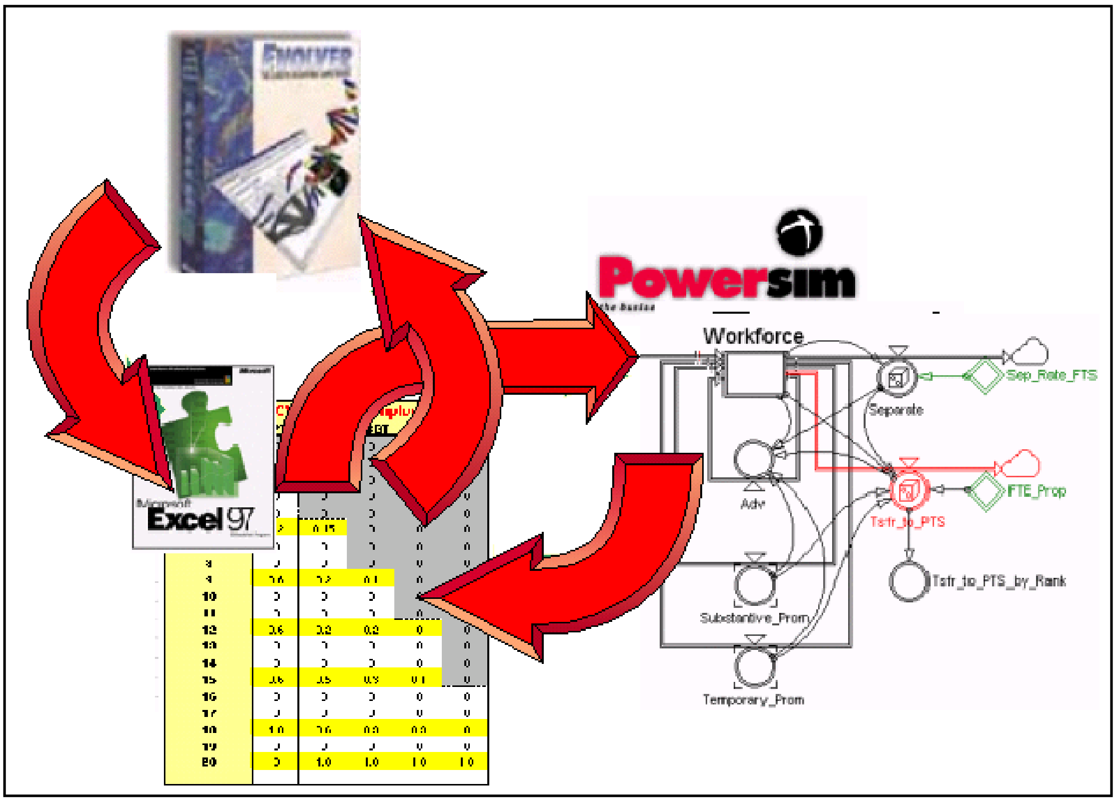


Figure 13: Optimisation data process flows

Optimisation Results.

Typically the optimisation process would result in a setting of the 'levers' which gave a 15% to 20% improvement on their initial 'considered judgement' position after 2,500 iterations. Running the system for a further 7,000 rarely improved the result by more that 1%.

Summary

This paper has outlined a powerful strategic enterprise employment simulation model. The strength of the model, its ability to track staff by rank, by time-in-rank, by years-of-service, is also a limitation because it results in about 50 'decision levers', each of which can have an infinite number of positions. Also, where different scenarios with different constraints are being compared, there is no prima facie basis for assuming the same 'ideal' initialisation settings.

The combining of genetic algorithm optimisation with the system dynamics model allows the automated identification of an 'optimum' initial setting of these levers (or at least a 'pretty good' starting point) for the different sets of constraints, from which the user can do 'what-if' analyses to understand the functioning of the system.

The optimisation process had the unexpected bonus of serving as a validation tool in that it ran very large numbers of 'extreme value' tests, occasionally produced aberrant results which, on reviewing, pointed to mistakes in logic or business rules.

Setting up the integration was no mean task. We look forward to testing the new Powersim Enterprise Kit with enhanced genetic algorithm capability to see whether it is capable of handling the complexity of large array models. If so, we will dispense with the integration process described in this paper.

---

[1] Wolstenholme, E.F., System Enquiry, A System Dynamics Approach, John Wiley and Sons, 1990, p.157.

[2] Coyle, R.G., The Use of Optimization Methods for Policy Design in a System Dynamics Model, System Dynamics Review, Vol 1, No 1, Summer 1985, p.81.

[3] Coyle, R.G., System Dynamics Modelling, A Practical Approach, Chapman and Hall, 1996, p.236.

[4] Sholtes, R.M., Optimising System Behaviour Using Genetic Algorithms, 1994 International System Dynamics Conference, Methodological and Technical Issues, p.246.

[5] Wolstenholme, E.F. and Al-Alusi, A.S., System Dynamics and Heuristic Optimisation in Defence Analysis, System Dynamics Review 3, No. 2, 1987, p.102.