INVITED

Event Driven and Continuous Modelling Systems : Update Review of Two Simulation Packages

Brian P Murphy
Department of Information Management & Marketing
University of Western Australia

Abstract We discuss recent powerful additions to two important simulation systems for PCs, **Extend**™ and **SimView**™ and emphasise via examples that these systems are capable of modelling real-world business, industry and management systems to any level of complexity, providing a comprehensive 'What If?' tool to aid strategic, tactical and operational planning in management. We discuss briefly their general facilities, and concentrate on new special features - the Traks modules of **SimView** and the clones and Report Writer of **Extend**.

1. INTRODUCTION

It is said that if the efficiency of or access to a tool is improved by a factor of 10, then quite new uses are found for it. If it is improved even further, a whole new rationale for its use will even appear. The chain-saw and its result, the modern ultra-destructive timber industry, is an unfortunate but instructive example. We discuss here additions to two newish (1990/1) simulation packages with this level of improvement factor, so that they have already started to massively redefine the users and usage of simulation techniques in business areas.

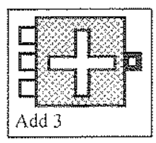
The programs of interest are **Extend** Version 3.1 [1995], and **SimView** Version 3.7 [1995]. Early versions of both have been discussed in some detail before by Murphy *et al.*, [1993], but recent extensions have considerably enhanced their powers. They are not labelled as 'simulation programs' by their authors, but as 'Dynamic Decision Support Tools'. The reason is that their major perceived market is the business manager rather than the scientist. This is a valid marketing target, not just because there are more (and better funded) businesses than scientists, or because scientists often are already accommodated by some earlier simulation program, but because these 'what if?' systems have powers to influence business and management decision processes in major ways, not possible by previous technologies, including even the modern powerful spreadsheet.

Since most business systems are about balancing demands (sales, profits) with resources (equipment, workers, funds), both under various levels of uncertainty of performance, simulation is a useful tool for studying the systems. Certain types of (large) business (mining, steel, railways, telephones) have been using simulation for decades, but their costs, mostly in terms of computer specialist staff required, have always been considered too high for smaller businesses. In our experience, the programs here discussed can cut these costs by 80-95%, by removing the need for expensive computer staff to actually build models, so that smaller businesses of many types now can benefit.

These programs are useful for scientists too, once scientists realise their particular virtue, that it is much easier to build/verify/modify a model in these very visual GUI systems than in any other, is relevant to them too, as the cost of building a model is usually a far more crucial matter than the cost of running the simulations. Computer time costs (on personal PCs) are now trivial, but salaries of computer specialists escalate, and the major aspect of the GUI programs is that they do not require computer specialist staff. So we are starting to see a new class of user - a professional in the business of interest who is computer literate, but not a programmer, as required by most older systems.

2. SOME PROGRAM ATTRIBUTES

To escape the need for programmer-level modellers, both these systems apply a similar approach. Visually identifiable icons, available from supplied libraries, provide the objects (e.g. queues and servers) which are placed on a worksheet, and agents and items (e.g. customers) which move through the model along defined paths through the objects, with which they interact (by waiting in queue, say). A model is thus constructed on the worksheet as a snapshot of the business process at a point in time, each block being in fact a subroutine, which computes certain outputs from the inputs at a time point. It is made to emulate action by calculations being repeated as 'simulation time' advances; this is analogous to a movie film, which is a fast-forward display of a series of still photographs, each taken in sequence a short time increment apart. Non-programmers can easily set up such models, though they do need to have some feeling for the concept of sequencing computations with action required. We find that our good students, with no programming experience at all, become quite competent in the mechanics after a 50 hour lecture and practical course, with equivalent own time work.



The **Extend** program is a little easier for beginners to get to grips with, as it supplies about 100 icons, which are called *blocks* to handle the sorts of objects and items needed, many at a very obvious level. e.g. the 'add three inputs' block here shown. **SimView** offers only 20 objects, but each has numerous options, so that the learning curve for **SimView** is initially steeper, but again, about 100 hours work delivers considerable skills. The blocks/objects are joined by paths showing the flow of the items. The fact that these blocks are a) general and often very powerful, b) well-tested (both products have been in the market some years) means that the modeller is not concerned with formal programming, but merely must fully understand the blocks and objects, and what they calculate and imply.

3. EXAMPLES: TRANSPORTATION PROBLEMS

The **SimView** examples below illustrates some of the general features of both programs. It is only possible to construct these model easily in **SimView** , because of the use of **SimView** 's unique *Traks* object, which can simulate the operation of any type of lines of railway track. It is of course possible to replicate this in other systems, with considerable difficulty, as it is a famous problem. **SimView**'s treatment looks sound (the algorithms are not published), and has been widely accepted, so it has a large advantage over all other systems in classic and special transportation problems.

3.1 A Railways and Port Example :

The first model (Fig 1) is relatively simple, but it was constructed for a very important study into rail facilities available for an Australian manufacturer. Product made at the plant is railed in bulk to the port, where it is stockpiled awaiting random ship arrivals. It can also be stockpiled at the plant, usually avoided as it involves double handling costs. Plant production is fairly constant, but there are shortfalls due to occasional breakdowns, and delays at the junction with the mainline traffic. The goal is to derive rules/schedules for train/ship despatches, given plant capacity and projected sales.
It thus addresses a long-term, strategic interest, and is a classic problem faced by many manufacturers.

The model is quickly constructed and the above questions can be answered by runs with various payloads, schedule schemes and rates of ship arrival. The necessary icons are obvious, and are joined by the appropriate paths. This model also shows some icons which are complete submodels, each made up of standard **SimView** objects. There can be a hierarchy of submodels (and they are called hierarchals in **Extend**), which means that models can be constructed by iterative refinement, progressing from outline to detail in a structured, test/validate-as-you-go, way.

We see in Figs. 2 - 4 some submodels of the system. These are all relatively simple, and show how iterative refinement enables us to build a complex model in a series of simple stages.

A major aspect of modelling in both **Extend** and **SimView** is that they have quite powerful animation. **SimView**'s is more explicit in the model above. As the simulation progresses, trains moves along the line, and ships move in and out of the port (there is one just about to leave the port in Fig. 1, lower left corner), and trains are unloaded in the submodels below.

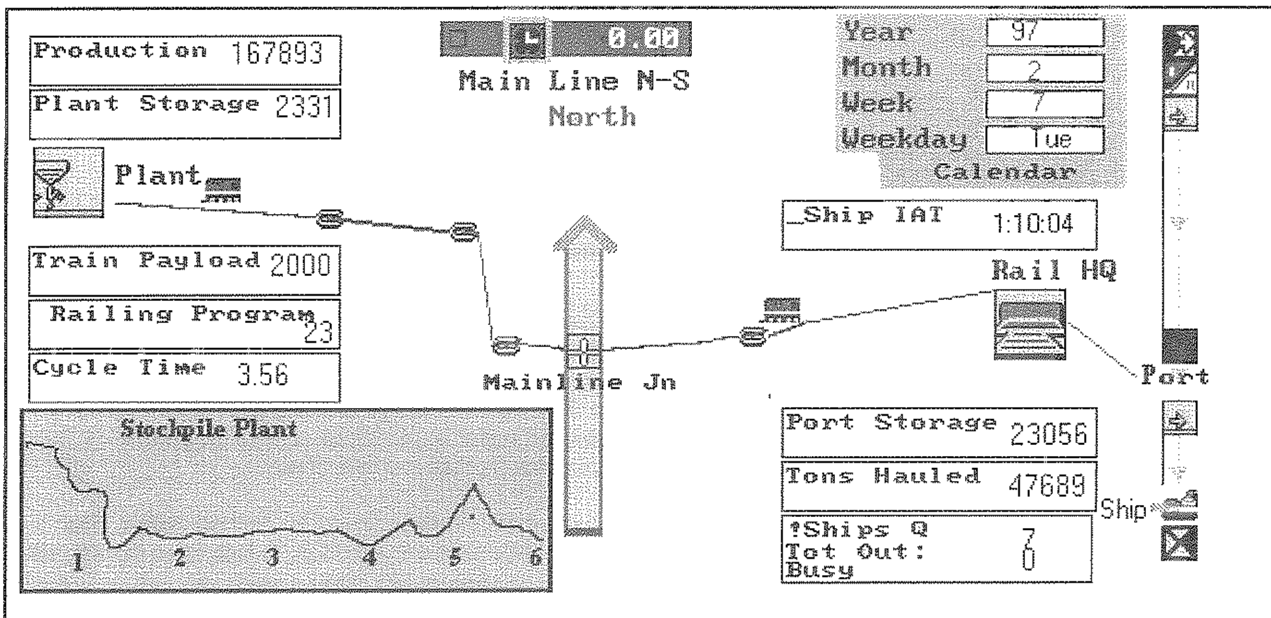


Fig. 1 Plant to Port via Train Model

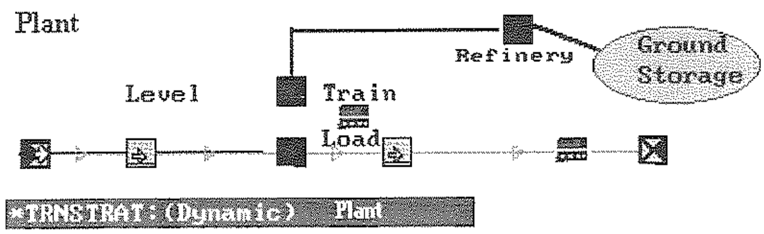


Fig. 2 At the Plant Submodel

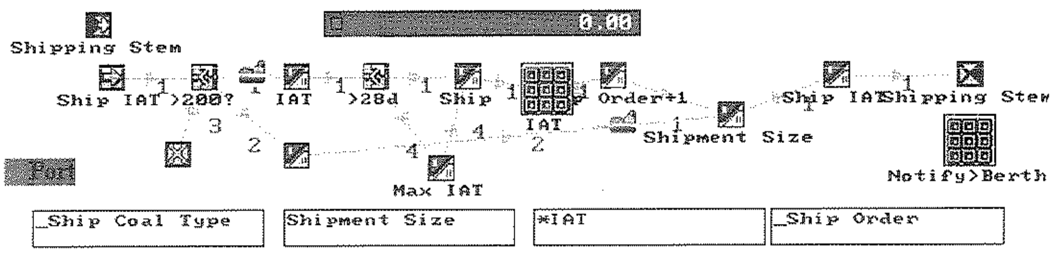


Fig. 3 At the Port Submodel

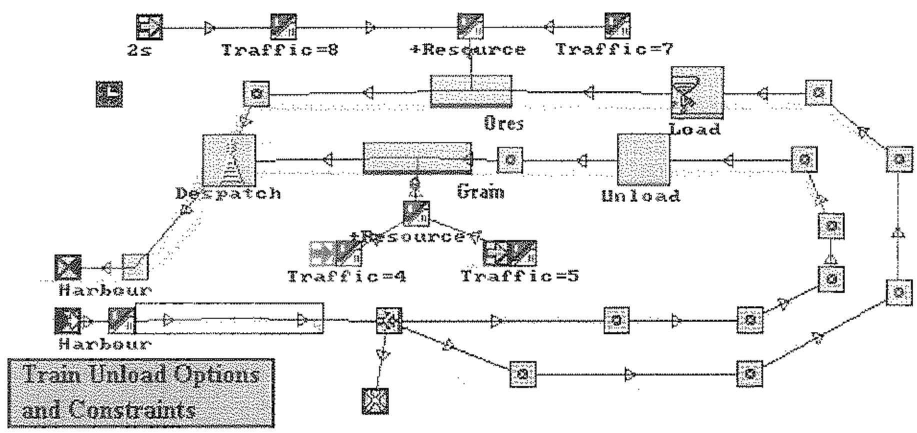


Fig. 4 At the Unloading Bays Submodel

This is not just cute - animation is a powerful debugging tool, but more importantly helps persons, not themselves modellers, to understand the operation, and assure themselves of its relevance, which in turn. builds confidence in the modeller and his results, something that those whose results are long tables of figures incomprehensible to many CEOs, no matter their virtue, have difficulty in obtaining (ask any statistician). Of course, the animation is time-expensive, and can be turned off when speed is required.

The investigation ultimately requires multiple runs with multiple parameters and switches (to allow say extra equipment/trains in some runs). A major recent facility offered by both these programs is an improved ability to set up and report on a series of runs with complex varied parameter options in a unified way.

Extend calls it their *Sensitivity Analysis* option; **SimView** has its *Instrumentation* concept. In both cases, inputs are set up in spreadsheet-like tables (which can also be directly imported from most spreadsheets); the user drives the program, which may or may not be hidden, selecting the input tables and schedules, and watches the results in output graphs and tables adjoining.

these under discussion can test schedules, and so give a nearly satisfactory solutions.

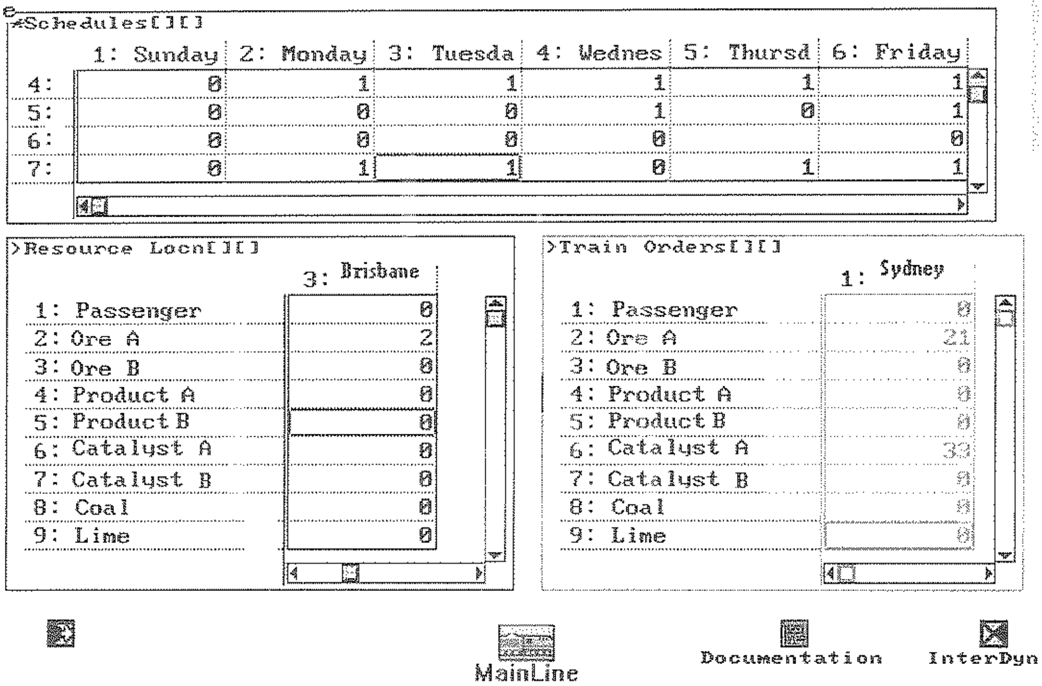


Fig. 5 Driving Runs by Instruments (Schedules, Tables)

The instrumentation concept of **Extend** exists in its facility for making clones of any object, graph or dialog box field, and its Notebook sheet, which is simply an area to place clones to create an in-out panel. **Extend**'s new Report-Writer utility further enhances this.

In most cases what managers are doing is trying to optimise a system over a range of inputs, and this approach helps to automate the searches needed. This semi-automated approach allowed by the Instrumentation setup is not an ideal 'goal-searching' procedure, which is of course not possible (ever?) in such a series of non-linear probabilistic equations and conditions, but it has been found quite useful.

Another transportation example is seen in the model below, which again is an extract from a real-world model of a national manufacturer supplying product to state retail branches and exporters. The purpose of the investigation is to refine the delivery scheduling process in terms of returns over transport costs. The need to construct schedules, which operate with some random failures components, is a classic problem, and non-soluble deterministically. But systems like

In this model the Instrumentation approach impinges on the main model itself - the output block of results is visible as model runs. Other instruments - truck schedules and retail branch demand tables can be easily called up to vi^w as the run proceeds, as in the Fig 6.

The Traks module is clearly a main feature of **SimView** , and unique, but it must be kept in mind that **SimView** is a genuine generic system, available for all discrete event work.

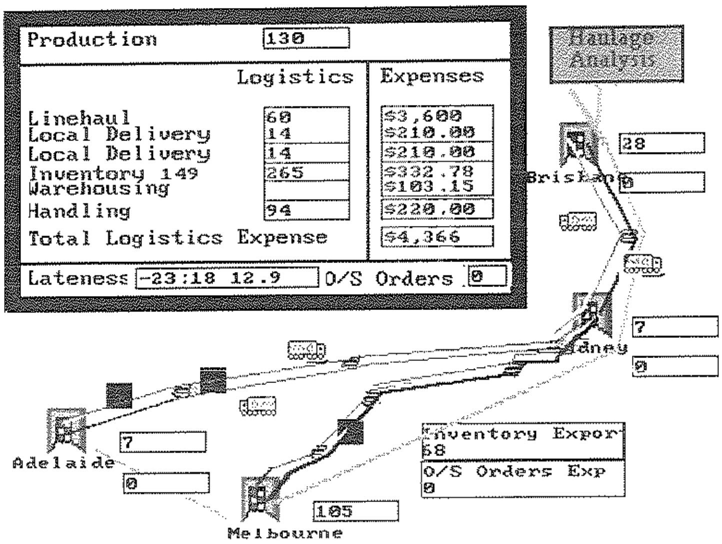


Fig 6 Trucking from Factory to the Retailers

4. A CONTINUOUS SYSTEM MODEL EXAMPLE

The **SimView** models above are all Discrete Event models - i.e. they are driven by events. The **Extend** program has extra blocks for Continuous Systems models. Fig. 7 shows a model for creating pseudo-histories of correlated rainfall leaching algae-sustaining nutrients phosphorous and nitrogen etc. into an estuary, in this case the Swan River of Perth, W.A. In this model we do not have events, or items flowing through the model, but quantities, i.e. numbers, flow from object to object, being transformed at each stage.

Rain is generated as a (random) quantity on a per week basis in the first section of the model (of which only parts are shown here), and the rain level determines how much of each nutrient is leached from the different types of land units (Section 2 of the model) and summed in the estuary (Section 3). All the numbers used up to this stage are products of much past research giving us records of nutrient release against rainfall and resultant flow of each river constituting the Swan Estuary Catchment area. In the next phase of the model, the nutrient is decremented by flushing with the ocean and mixing through the watercolumn. This too has relatively sound records to support it.

In the third phase of the model we attempt to plot the weekly use of the nutrients by the various algae. This is much more speculative, for while records on algae growth are extensive, there is less understanding and fewer useful records of how they hibernate and die. This matter is being addressed seriously by the various water monitoring bodies, but meanwhile the model gives some help in the management process of the problem as it exists now.

It was constructed and validated in a few days, as against months needed for an earlier almost similar job in the Fortran language.

Continuous models are more common in environmental and ecological research, and this makes **Extend** very useful in these areas. I would even say predominant over all others, though one can always fudge up a DE model to look continuous.

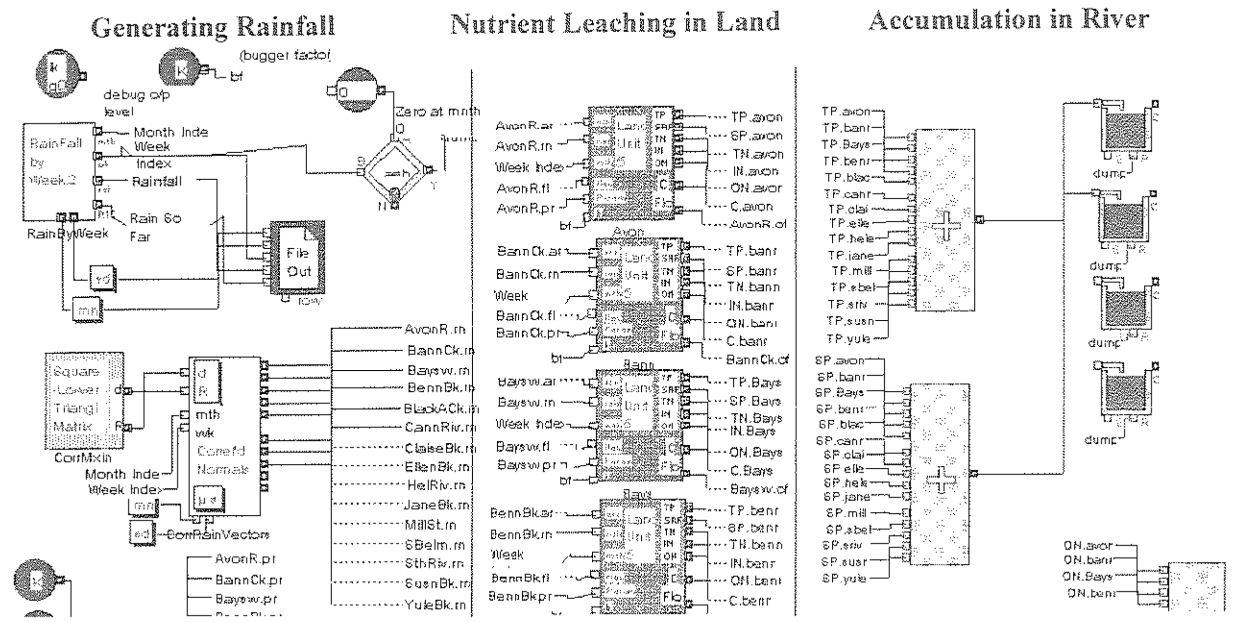


Fig.7 Swan River Nutrient Estimation Model

5. CONCLUSION

Through these examples we see programs which are powerful enough, we believe, to handle very large simulation problems. They are relatively easy to learn by competent persons who are not computer programmers, and this gives them a large advantage over competitors, because it goes with greatly decreased costs, in our experience by as much as 95%. They are also visually explicit, and this with judicious use of animation creates confidence in believing their results. Finally, we can recommend both. **Extend** is easy to get started, quite powerful, good for any type of model but particularly well suited to continuous models. **SimView** has spectacular graphics, and much power in on the side computations (e.g. for tracing cash flows simultaneously with the action); it is most impressive in discrete event models, and has a superb feature, and maybe the only satisfactory system, for handling transport problems, particularly where single tracks are concerned.

6. REFERENCES

1 **Extend™** (1988,1995): From Imagine That, Inc., San Jose, CA 95115 (Cost approx. $AUD2000)

2 **SimView™** (1990,1995): From InterDynamics Ltd., Inglewood, South Australia (Cost approx. $AUD8000)

3 Murphy, B.P. & C. M. Randolph, Simulation Packages Today : New User Classes and Order-of-Magnitude Breakthroughs in Costs, **Computational Statistics and Data Analysis, 16,** 4, 471-9, 1993

4 Murphy, B.P., & D.M. Deeley, New PC-based Simulation Packages for More Efficient Development of Statistical Models, **Proc. Intn'l. Congress. on Modelling and Simulation** (joint hosts IMACS - Simulation Soc. Aust.), Uniprint, Univ. West Australia, 1407-11, 1993