

https://dspace.org/

# A System Dynamics model for evaluating the performance of a Technical Support Organisation

**Williams, Hansell Hatfield**

**2007-02-15T11:04:22Z**
http://hdl.handle.net/10539/2000

Downloaded from a DSpace repository. https://dspace.org/

# A System Dynamics model for evaluating the performance of a Technical Support Organisation

**Hansell Hatfield Williams**

**A research report submitted to the Faculty of Engineering and the Built Environment, University of Witwatersrand, Johannesburg, in partial fulfillment of the requirements for the degree of Master of Science in Engineering.**

**Johannesburg, July 2006**

## DECLARATION

**I declare that this research report is my own, unaided work. It is being submitted for the Degree of Master of Science at the University of Witwatersrand, Johannesburg, South Africa. It has not been submitted before for any degree or examination at any other University.**

**Signed this _____ day of _____ 2006**

_____

**Hansell Hatfield Williams**

## Abstract

Technical support organisations (TSOs) and service groups are becoming increasingly important in the upkeep and maintenance of the highly sophisticated and optimised automation equipment and systems that drive production in the mining and manufacturing industries. This study presents a methodology for designing and analysing the performance of a TSO, with a view to ongoing optimisation. The method of approach is the 'System Dynamics' (SD) simulation technique, which has at its core a mathematical model of the organisational unit, and which attempts to model its behaviour over time. The organisational design is based on Integrated Logistics Support (ILS) principles. It was found that the System Dynamics modelling approach was useful for a technical support organisation. For managers and organisational strategists, the quantitative ability of SD provides an intuitive and practical way to evaluate the impact of different structural and operational scenarios in response to ever-demanding business imperatives.

## Acknowledgements

I wish to thank the following people for their contribution to this M.Sc. research project:

Professor Brian Wigdorowitz, School of Electrical and Information Engineering, University of Witwatersrand, for his patience and invaluable assistance throughout.

My wife Jenny and our family, for their ongoing support and encouragement.

The De Beers company, for their support, and as the source for project data.

My colleagues at De Beers, for peer reviewing and scrutinizing of the material.

Cathy Snow, for proof-reading and editing.

# CONTENTS


| | Page |
|---|---|
| **DECLARATION** | **2** |
| **ABSTRACT** | **3** |
| **ACKNOWLEDGEMENTS** | **4** |
| **CONTENTS** | **5** |
| **LIST OF FIGURES** | **7** |
| **LIST OF TABLES** | **8** |
| **Chapter 1 – Introduction** | **9** |
| **Chapter 2 – The Nature and Challenges of Technical Support** | **11** |
| **Chapter 3 – Managing TSO Performance** | **13** |
| 3.1 Literature survey | 14 |
| **Chapter 4 – Organisational Modelling and the System Dynamics (SD) Method** | **15** |
| 4.1 The need for organizational modelling | 15 |
| 4.2 The Systems dynamic technique | 15 |
| 4.3 Structure determines behaviour | 16 |
| 4.4 Systems Dynamics, Systems Thinking, and Case Studies | 17 |
| 4.5 Systems Dynamics in detail | 19 |
| 4.6 Software tools | 21 |
| 4.7 Application example – personal savings model | 21 |
| 4.8 Formulas behind the SD model elements | 23 |
| 4.9 Savings model refinement | 24 |
| 4.10 Other applications of System dynamics | 25 |
| 4.11 A summary of the benefits and drawbacks of SD | 26 |

**Chapter 5 – How the TSO Model was Developed** **30**
5.1 SD model development process 30
5.2 Drawing on the Support Logistics discipline 32
5.3 Validity of the model 33
5.4 Modelling style 34

**Chapter 6 – Full Description of the TSO Model** **35**
6.1 Introduction 35
6.2 Sector overview 36
6.3 Data gathered for concept design 37
6.4 Implemented model structure 41
6.5 Detailed description of model sectors 43

**Chapter 7 – Model Testing and Results** **49**
7.1 User interface control panel 49
7.2 Model validation 50

**Chapter 8 – Conclusions and Further Work** **58**
8.1 Suggestions for refinements and further work 58

**References** **60**

**Bibliography** **61**

**Appendix A – Glossary of terms** **63**

**Appendix B – The 'iThink' software modelling tool.** **64**

**Appendix C – Model data** **66**

**Appendix D – Model equations** **72**

## LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 4.1 | Basic Stock and Flow elements | 16 |
| 4.2 | System Thinking's Casual Loops | 18 |
| 4.3 | Stock and Flow diagram | 19 |
| 4.4 | A simple application – Personal Savings system | 21 |
| 4.5 | Graphical input structure representing 'Emergencies' | 23 |
| 4.6 | Savings model refined | 24 |
| 5.1 | TSO model development process | 30 |
| 5.2 | Equivalent representations | 34 |
| 6.1 | TSO model overview | 36 |
| 6.2 | Example of an input data component: 'Call-desk availability' | 41 |
| 6.3 | Documentation | 42 |
| 6.4 | Decision logic in iThink | 43 |
| 6.5 | Model overview by sector | 44 |
| 6.6 | Problem sources | 45 |
| 6.7 | Productivity, motivation, burnout (PMB) | 45 |
| 6.8 | Central support personnel | 46 |
| 6.9 | Remote support PMB | 46 |
| 6.10 | Central infrastructure | 47 |
| 6.11 | Remote infrastructure | 48 |
| 7.1 | Simulation dashboard | 49 |
| 7.2 | Steady state conditions | 51 |
| 7.3 | Rework level at 20% | 53 |
| 7.4 | Support systems failure | 54 |
| 7.5 | Task overload | 55 |

## LIST OF TABLES


| Table | | Page |
|---|---|---|
| 4.1 | Model elements | 20 |
| 4.2 | A detailed exposition of the elements of Fig. 4.4 | 22 |
| 6.1 | Data collected for model development | 38 |
| 7.1 | Sensitivity analysis: Varying adjustable parameters | 57 |

# Chapter 1 – Introduction

## Introduction

Technical support organisations (TSOs) and service groups are becoming increasingly important in the upkeep and maintenance of the highly sophisticated and optimised automation equipment and systems that underpin production in the mining and manufacturing industries.

Efficient management and the ability to plan ahead are key factors in light of the pressures on cost-of-services. Current methods of operational management are driven by helpdesk data, spreadsheets and databases that track outstanding calls, deployment of personnel, spares movement, etc. Importantly, there is also the dependence on an experienced manager to interpret the data and to intervene when undesirable trends occur. However, these methods do not provide any insight into the non-financial aspects of the service business. Issues such as staff motivation, quality of work and rework, the impact of new trainees from a training programme, customer perceptions, and others, are difficult to incorporate into traditional management methods. As situations become complex and the number of variables increase, even experienced managers are challenged as the limits of traditional methods are reached.

In order to meet these challenges, the need arises for a more sophisticated method of modelling organisational systems that is able to incorporate all factors – data and human. 'Systems dynamics' modeling provides such a platform. The model would overcome the shortcomings of traditional spreadsheet and trending methods by providing an integrated simulation of the underlying behaviours that influence the TSO outcomes over time. Not only would the model become a valuable asset in a manager's inventory, but it would facilitate understanding of the 'systems' nature of the business by its personnel. Also, as the simulator is constantly refined, it becomes an increasingly useful tool for the efficient daily running of a complex unit.

Another use of the model would be as a training aid for new managers, allowing them to take on their new roles with a deeper level of understanding, thereby cutting down on some of the unnecessary 'noise' of the learning curve. It also provides the ability to extrapolate into the future, using the bases of organisational structure, policies and expected and observed behaviours.

In the current and future economic climate, with its emphasis on low cost of services, sustainability and continued growth for stakeholders, such tools are likely to find much favour with operational and business managers alike.

The purpose of this study is to develop an idealised TSO structure, and model it using the System Dynamics method. Sample simulation analyses are undertaken,

and conclusions are drawn as to the applicability of the technique to technical support organisations.

**The research report is arranged as follows:**

Chapter 2 explores the challenges and nature of Technical Support.

Chapter 3 reviews the current methods of managing TSO performance.

Chapter 4 introduces the System Dynamics method and its application to a TSO.

Chapter 5 describes the process through which the TSO structure was developed, followed by a detailed description of the model in Chapter 6.

Chapter 7 deals with model evaluation and testing, and

Chapter 8 draws conclusions and provides recommendations for further improvements and future work.

## Chapter 2 – The nature and challenges of Technical Support

The purpose of a TSO is to provide a corrective, preventative and advisory service for technical equipment and systems for production-oriented operations. Variability in demand is endemic. A further goal is to deliver such services at the lowest cost, at or above quality expectations, while maintaining low turnover in the human resource pool so that a knowledge repository can be developed.

A typical TSO would comprise a skilled and experienced human resource component, and systems or facilities and work processes to support it. In the mining setting under study, the support system comprises an on-mine TSO, supported by a central or head-office TSO. Differences between the teams are levels of expertise, types of problems dealt with and the response-time factor. The on-site team should respond rapidly and solve 80% of the problems. If too complex a problem arises, the local team will typically implement contingency plans or work-arounds, allowing the central team more time to resolve the (more difficult) problem. However, it is anticipated that on occasion no work-arounds are possible and if production is affected, the central team has to respond on the same basis as if it were the onsite team.

Under normal circumstances, the TSO would be in a good position to meet normal demands. However, a confluence of events could create service demand levels beyond its capacity. Unless such situations are handled with foresight, preparedness and firm management skill, the TSO could fail to meet expectations.

This highly variable volume of service requests is typically dealt with by a fixed resource pool with varying levels of expertise over a spectrum of skills; this pool has to maintain a minimum standard of service quality. Coupled with this is the reality of a constantly-changing environment: technology advancements and upgrades, support staff movements and availability, configuration changes at the target sites due to modifications to plant and equipment, new projects coming on stream, and the uptime/downtime of support systems such as networks and test

equipment. The increasing sophistication of user requirements drives the need for higher output and productivity, typically at the expectation of no cost increases.

With regard to the critical human resource component, it is an accepted fact in the high-technology service field that the highest levels of experience, skill and knowledge are required for faultfinding and fault correction. In contrast, a system developer will proceed according to a design and standards, generally using a narrow set of skills and under far less time-pressure. In problem-solving, the broadest base of knowledge is required to deal not only with hard technical issues, but wider system influences – other components linked to the system that may have an indirect impact. User and soft issues, often-extreme time-pressures, and environmental impacts pose further challenges. A mitigating factor in the increasing trend of technical sophistication has been the growing 'design for maintenance' focus over the years. Extensive use of self-diagnostics and modular design allow for throw-away or rapid-replacement components. Highly-productive technical support specialists are scarce resources: they typically have many years' experience and would ideally come from a product development environment. This background would ensure the necessary deep knowledge and insights into the systems under their charge. The dearth of skills is exacerbated in the case of remote-mining, as highly qualified individuals prefer to work in the cities for a better quality of life. One of the few ways to attract and keep skills in remote places is through financial incentives. This brings us back full-circle to the need for good management tools and support systems in order to plan, design and optimally run costly support services.

Further challenges faced by TSOs are shorter product life-cycles and more sophisticated products; higher productivity is required from fewer resources, and multi-skilling is essential as support specialists also need to perform project duties. Additionally, haphazard training and skills development are the result of variable demands on the staff complement.

## Chapter 3 – Managing TSO performance

Spreadsheets and reports are the typical and prevalent methods of tracking and analysing support group performance over time. Since most modern organisations utilise helpdesk or call management systems, support statistical data is easy to come by (HEAT, Footprints). Typical analyses and reports would reflect: incoming calls per period (day/week/month), calls closed (solved) per period, current number of calls outstanding and personnel utilisation, among others. However, the focus is on outcomes, and does not provide deeper insight into the underlying dynamics and cannot easily show interactivity between factors. Reports are discrete in nature, reflecting only the current status, and at best showing historical trends. A longer-term, continuous assessment of operational performance requires a model of the system, which when fed with parameter-type data, can plot out future projections of key variables. With traditional methods there are no consistent ways to introduce outside factors, such as how new projects affect call load, or how staff 'burn-out' causes a reduction in call closing rate.

More traditionally, the strategic planning and optimising of a TSO fell to an experienced, hands-on support operations manager, who could examine performance data and utilise his mental models gained from years of experience to predict crisis situations and the possible impact of external factors. However, as situations become more complex and the group becomes larger, information overload diminishes his ability to navigate ahead with foresight, especially in the management and balance of the nine or ten Integrated Logistics Support (ILS) sub-disciplines that constitute support organisations. (ILS, Chapter 5.2).

Forrester (1991) proposes that information about organisational performance be classified as follows: mental database, written database, and numerical database, and that information content, i.e. policies, structures, and behaviours, *decreases* from mental to numerical databases. Since organisational models contain data

from all three databases, to simply focus on performance reports and time-series data - a typical approach - would produce a very limited view of TSO performance. All three types of information are required to build an adequate model with which to study organisational behaviour.

### 3.1 Literature Survey

The System Dynamics technique itself is well represented in the literature, with a significant bibliography having developed over some 35 years [See Bibliography], ably reviewed by founder Forrester in *System Dynamics and the Lessons of 35 years* (1991).

However, very little appears in the literature regarding a service delivery model for a TSO. This is surprising, considering the explosive demand for end-user support on the back of ever-advancing and ever-'upgrading' IT technologies. The focus appears rather to be on the technologies in vogue: help-desk systems, web-based support, and remote support via the Internet or networks, as typified by (Foo, 1999). These topics fall into the ILS area of 'Facilities' (ILS, Chapter 5.2) and may be essential cogs in the TSO structure, but with such a narrow focus they do not provide a time-dynamic view of TSO performance, such as in the technique proposed here.

Oliva and Sterman (2001) have considered a specific aspect of service delivery - 'service quality' - using System Dynamics modelling. Their proposal addresses deeper-level dynamics concerning the effects on service quality due to unwise working-time practices. In the Integrated Logistics Support framework which is used as a basis for developing the TSO, 'service quality' falls into the category of 'Organisational planning', and is not specifically addressed in this study. However, the TSO model, through the use of System Dynamics, allows any underlying aspect such as service quality to be further developed, and be incorporated with ease.

# Chapter 4 – Organisational Modelling and the System Dynamics (SD) Method

## 4.1 The need for organisational modelling

*Definition of a system: 'An entity that maintains its existence through the mutual interaction of its parts'.*

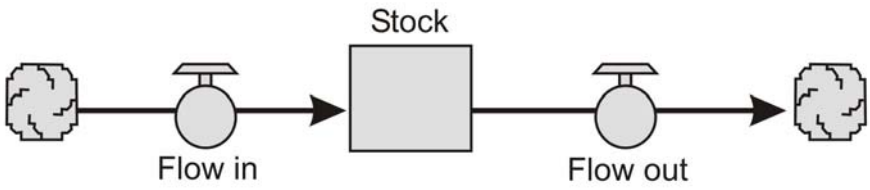
Most organisations now accept that, for significant and sustainable operational improvements to occur, they need to consider not only the discrete contributions that individuals and organisations can make, but how the whole system works together. Senge, in *The Fifth Discipline* (1990), popularised 'Systems Thinking', which aims to sensitise organisations to the need for the 'big picture' approach, and proposed causal loops as a way to depict large interconnected systems.

In the context of a TSO, the group will have as a minimum human resources and support systems. Each of these is a subsystem, all of which together form a complex TSO 'system'. Interactions now easily grow more complex and non-intuitive in the time-domain, and here the System Dynamics method, also known as 'stock and flow' provides a means by which the key stocks in the system can be identified, analysed quantitatively, and used for future planning and organisational change management.

## 4.2 The System Dynamics technique

The System Dynamics (SD) discipline was developed by Jay W Forrester (1967) in the 1960s for describing and analysing the time-dependent behaviour, or dynamics of 'systems'.

Forrester's 'stock and flow' technique is central to System Dynamics and uses the concepts of a 'stock', or level or quantity of something, and the flow rate into, and out of such a stock, as per Fig. 4.1.




**[At any time: $S(t) = \int$ (Flow in – Flow out).dt)]**

**Fig. 4.1** Basic Stock and Flow elements

The difference between the rates of inflow and outflow determines the stock level at any point in time. Forrester's assertion was that a wide range of physical and non-physical processes such as organisations could be modelled by considering the levels of material or non-material entities, and how they changed in time. In organisations, stocks could be physical, such as finances, or non-physical, such as people's stress. An intuitive diagrammatic technique was developed alongside the mathematical underpinnings. Stocks and flows are depicted as tanks and valves controlling the inflow and outflow of material, complemented by 'converters' to change between materials. 'Connectors' are information flows and link the entities together.

### 4.3 Structure determines behaviour

Forrester (1991) contends that the structure of a system has a fundamental impact on the behaviours observed within that structure, over time. In business, the intuition or 'common sense' applied by managers or designers relates to their 'mental models', or the understandings and insights they have gained of the business, and developed with experience over time. Organisational structures developed solely from mental models pre-determines certain longer term

behaviours, which in the future may be unintended and undesirable, or at best, at odds with its operating environment. The SD approach attempts to eliminate the 'guesswork' using a technical simulation. This approach exposes the nature of time-related behaviours for scrutiny and understanding, but in a mathematically rigorous way.

### 4.4 System Dynamics, Systems Thinking, and Case Studies

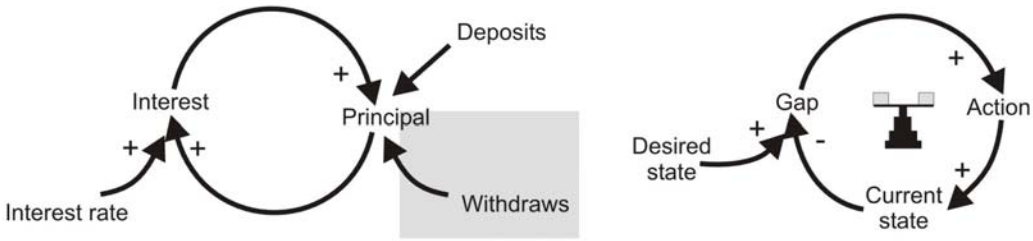
Much debate rages over the relative positioning of the various methods to model systems. Forrester (1992) makes a compelling argument for the SD approach to be comprehensive enough to address all the necessary steps in a systems modelling process. The steps proposed are:

Step 1: Conceptual phase or description of the system, where situational information is gathered and organised;
Step 2: Convert concepts into model equations;
Step 3: Run the simulation and analyse behaviour;
Step 4: Modify the structure and policies (behaviours or parameters) to improve performance;
Step 5: Educate, review, debate (cycle through steps 3-5);
Step 6: Implement changes in the real world.

The traditional systems modelling and analysis approach is achieved through case studies, a non-technical, unstructured method. Case studies are conducted with groups of people who have a wide range of organisational experience. Problem-solving and analysis is achieved through debate and discussion.

The 'Systems thinking' approach (Senge, 1990) also uses a diagrammatic technique known as 'causal loop' analysis, as shown in the examples in Fig. 4.2 below. Causal loops attempt to identify the entities in a system and their relative impact on each other. Senge proposes two basic structures, 'compounding' and

'regulating' loops, which, in myriad combination could form complex, interacting organisational or social systems.




***Compounding or reinforcing loop*** ***Regulating loop***

**Fig. 4.2**: System Thinking's Causal Loops

Forrester contends that much as these 'soft' approaches can correctly identify organisational structure and behaviour (policy) and can stimulate possible solutions, they do not have a quantitative next step to evaluate such solutions 'dynamically' over time. Furthermore, without a mathematical underpinning, essential simulation tasks such as 'sensitivity analysis' cannot be done.

It is proposed that these techniques belong in the initial conceptual phase (Step 1), but should then be followed by the more rigorous SD approach for the subsequent steps. The approaches are essentially complementary and it has been found that the Systems Thinking causal loop method is useful for teaching about systems in general and also for presenting an overview of an SD model.

### 4.5 System Dynamics in detail

In a typical Stock and Flow (SF) diagram, each stock represents a mathematical 'order', i.e. a diagram linking 3 stocks would be a 3rd-order linear or non-linear system.

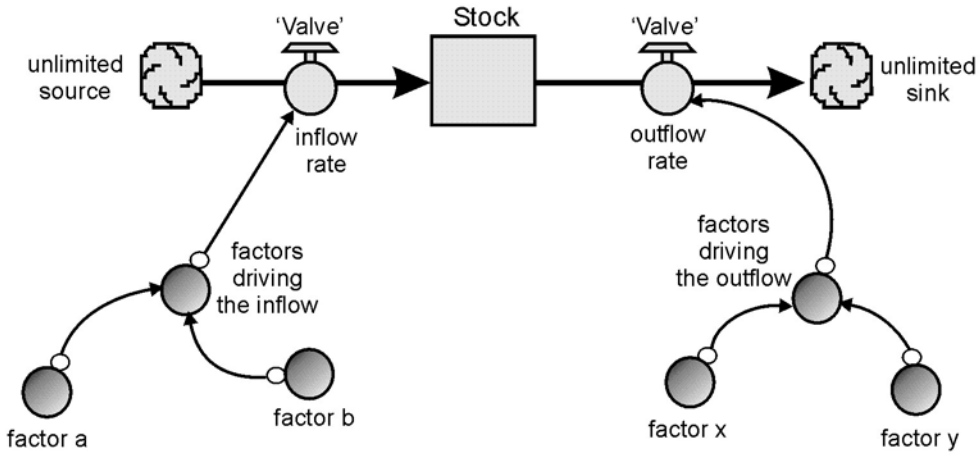



**Fig. 4.3**: Stock and Flow diagram

The concept of information or knowledge flow is introduced through 'converters' and 'connectors'. In essence, Fig. 4.3 can be described as follows:

"The combination of informational factors 'a' and 'b' influences the flow rate of 'material', which increase, the amount of such material in the stock. Conversely, factors 'x' and 'y' control the exit rate of the material out the stock."

Table 4.1 below describes the above model elements in detail.

**Table 4.1**: Model elements

| ***Model element*** | ***SD nomenclature or symbol*** | ***Description*** |
|---|---|---|
| Unlimited source | Source cloud. | Source of the material or main variable. It is assumed that the source of 'material' (for example, water, motivation, quality), is unaffected by the behaviour or demands of the model in question. |
| Double line with arrow | Material flow indicator. | Indicates the flow of material that accumulates in the stock, or depletes it. |
| Single line with arrow | Connector. | Connectors connect model elements, conveying information as manipulated by a converter. |
| Inflow rate, outflow rate | Flow. | A device for controlling the flow-rate into the stock, as influenced by the connected elements. Flow can be bidirectional. |
| Accumulator | Stock. | The integral of the difference between inflows and outflows, over time. |
| Factors driving the inflow, outflow | Converters. | Constants, or mathematical combinations of incoming factors (also converters). |
| Unlimited sink | Sink cloud. | The destination of the 'material' (for example, water, motivation, quality), is unaffected by the behaviour or demands of the model in question. |

### 4.6 Software tools

The SD method has been implemented in software tools such as 'iThink' (Appendix B), 'Vensim' and 'Powersim'. Graphical interfaces and various enhancements to the basic SD elements ease the model constructor's task.

### 4.7 Application example – Personal Savings model

Figure 4.4 depicts a simple SD example where a personal savings system is modelled. Included in the model are the typical banking concepts of interest rate and the savings account that accumulates funds on a compound basis. However, added to the model are the 'real-world' or people-oriented factors such as perception of personal wealth, emergency spending, and long-term financial plans. For example the 'perceived wealth' item is shown as a 'converter', meaning that the savings information (amount in the bank account), is multiplied by some factor, and converted to a quantity with similar units and value range as the 'spending' converter that collects all the spending-type factors. All right-hand side factors then drive the spending-rate valve, which effectively reduces the amount of savings.

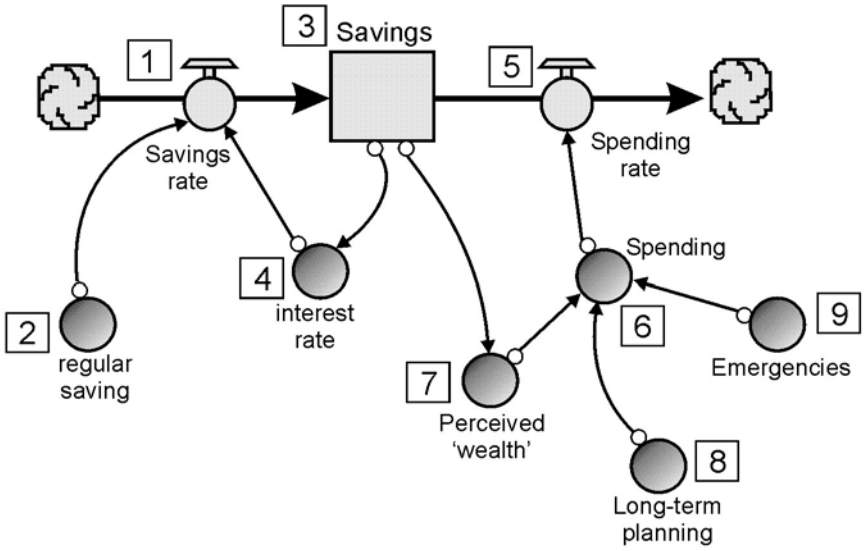



**Fig. 4.4**: A simple application – Personal Savings system

**Table 4.2**: A detailed exposition of the elements of Fig. 4.4.

| ***Model variable*** | ***SD entity*** | ***Description*** |
|---|---|---|
| 1. Savings rate | Flow | The rate at which the 'Savings' accumulator is increasing (i.e. rands per unit time). |
| 2. Regular savings | Converter | Regular amounts flowing in, influencing the rate of savings. |
| 3. Savings | Stock | Savings accumulator. |
| 4. Interest rate | Converter | An amount influencing savings rate, dependent on the current value of savings, and the bank's rate of interest. (which could change over time). |
| 5. Spending rate | Flow | The rate at which the 'Savings' accumulator is *decreasing* (i.e. rands per unit time). |
| 6. Spending | Converter | A combination of the factors influencing spending, normalised to match the units, time and value ranges of the 'spending rate' flow variable. (a summer in this case.) |
| 7. Perceived 'wealth' | Converter | A factor introduced to model a person's propensity to spend more, in general, depending on the value of savings. |
| 8. Long-term planning | Converter | A savings-reducing factor to simulate amounts planned to be expended from time to time (discrete occurrences). |
| 9. Emergencies | Converter | A savings-reducing factor to simulate the occurrences of discrete amounts of unforeseen spending. For long-term scenario planning, a random function could be used. In a real-time model which is updated on an ongoing basis, any emergency expenditure would be entered as and when it occurs. |

### 4.8 Formulas behind the SD model elements

In order to produce a quantitative output, all model elements are underwritten by formulae. For example, the stock element requires the use of an integral formula, of the form:

Savings(t) = Savings(t - Δt) + (Savings_rate - Spending_rate) * Δt [where Δt is the calculation time interval]

INIT Savings = 1000, [initial value for the stock]

Software tools such as 'iThink' provide a user-friendly and empowering way to build SD models, through the use of interactive graphics, self-generating formulae and a host of display and ease-of-use features. (Appendix B provides a description of 'iThink'.)

All inputs (or 'behaviours') can be introduced as graphs or tables of data, either as time-dependent, or linear or non-linear functions. For example, emergency outflows in real life would occur as discrete values over the time-period in question. This type of input is introduced into the model as in Fig. 4.5.

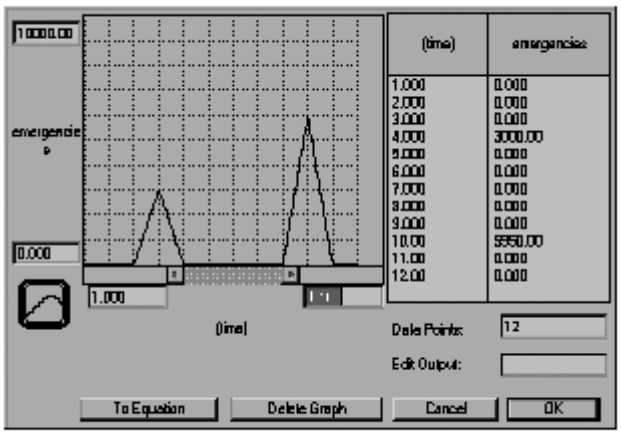



**Fig. 4.5**: Graphical input structure representing 'Emergencies'.

This feature not only simplifies the task of model building, it materially contributes to greater creativity and allows modellers to address more qualitative aspects than would normally be the case. The ability of SD to allow extensive exploration of the qualitative dimension makes it a compelling tool for the simulation of real-life situations.

### 4.9 Savings model refinement

Since SD models are concerned with changes over time, and since human factors are very likely to change over time, the 'perceived wealth' factor could easily be expanded into a separate stock and flow diagram, with a link from the new stock to the 'spending' converter. This sub-model would attempt to model the waxing and waning of a person's 'perceived wealth', by further considering the reasons for its increase and decrease. In this case, the bank's - and other vendors' – increased attention come into play, in addition to the influences that come with personal 'asset accumulation'.

Fig. 4.6 presents this refinement to illustrate the ease of extension and modification of the model, as would typically occur in Steps 3-5 of the Forrester modelling process (Section 4.4).

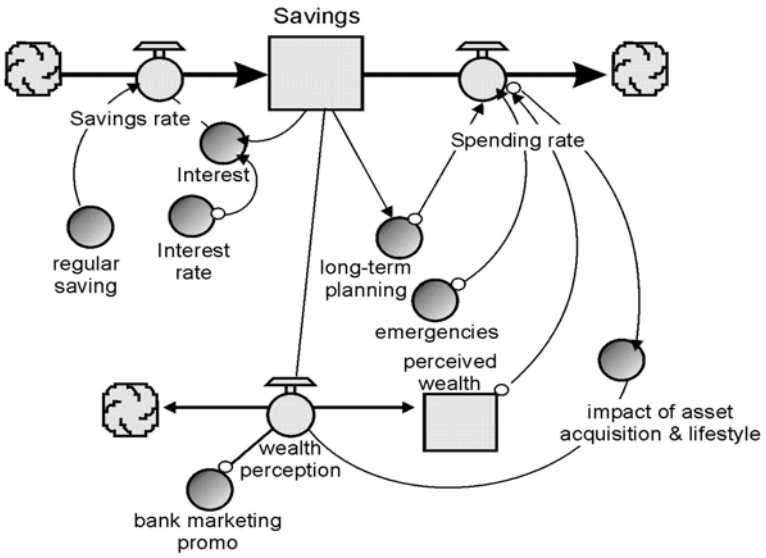



**Fig. 4.6**: Savings model refined

**Note: Bi-directional flow.**

A bi-directional flow element is used when the combination of the driving forces of the flow could vary positively or negatively.

### 4.10 Other applications of System Dynamics

The literature covers a wealth of real-world applications, from business cycle predictions through drug abuse models, to the simulation of Shakespeare's 'Hamlet'. Some more conventional examples are listed below.

**Dynamic Balanced scorecard**:
Stock and flow models are used for each of the scorecard dimensions – Financial, Customer, Internal Business process, and Learning and Growth. The graphical links between elements show the inherent interconnectivity between the underlying elements. [iThink v 7.0.3 documentation]

**Supply chain dynamics**:
The effects of multiple reservoirs, variable demand and internodal delays make for complex time-domain behaviour, ideally suited to SD modelling. [iThink v 7.0.3 documentation]

**Resource allocation in business generation and client servicing**:
In situations where an organisation has multiple priorities to balance, SD analysis highlights the impacts of headcount levels, experience levels and the time delays in the bidding and winning processes and recruitment delays to counter attrition. [iThink v 7.0.3 documentation]

**Financial resourcing processes**:
A model could identify the key elements associated with a company's financial cash-flow and how it would influence marketing budgets and expenditure, for

example. Spending level is controlled by factors such as liquidity, advertising spend, months' coverage, etc. [iThink v 7.0.3 documentation]

**Macro-economic models**:
Large models which attempt to predict business cycles, and have confirmed phenomena such as the 50-year 'economic long wave', or Kondratieff cycle. [Kondratieff, 1984]

### 4.11 A Summary of the Benefits and Drawbacks of SD Modelling

No matter how promising and appropriate a technique may appear for an application, an objective view is required. The following comments highlight some of the pros and cons of the SD technique.

***Benefits***

**First-principle modelling**: The technique provides for a fundamental modelling approach rather than a black-box method, such as neural networks. Provides insights and fuller understanding of the total environment.

**Manage levels of complexity:** Complexity level is chosen by the analyst, and typically starts out with a few, higher-level concepts, which can be progressively decomposed into lower level detail as domain knowledge grows and further data is collected.

**Input-data flexibility:** Real-world input data in various forms can be applied to the system, especially with SD tools such as 'iThink'. This data can be in the form of time-series, graphical or tabular data, behavioural functions, and discontinuous data, etc.

**Graphical metaphor for ease of visualisation:** The graphical elements are intuitive and simple and facilitate group interaction and knowledge sharing.

**Group workshops and team motivation:** The inherent graphical nature of SD is very well suited to group interaction and 'workshop' sessions.

**Teaching and training tool:** Similarly, due to the graphical metaphor, it is effective as a training tool, in teaching both new starters and experienced service personnel about the dynamics of a TSO. New starts will benefit from exposure to the explicit connectedness of the service environment and the key determinants for success, and experienced practitioners could have their mental models and 'gut-feel' confirmed. Simulators can also function as management training 'flight simulators'.

**Simulation:** Tools such as iThink naturally extend the modelling process to live simulation, and show stock and flow dynamics in terms of time-based graphs or tabular data. The control panel or dashboard approach makes interaction with the tool stimulating and 'fun', again beneficial for teaching or team building.

**As operational tool:** User interfaces provided by software implementations allow simulators to be used as management 'dashboards', where model-data is updated with real data on a periodic basis.

**What-if and sensitivity analysis:** Inherent in quantitative simulation is the ability to manipulate key parameters, and evaluate the behaviour of important stocks.

**Data set optimisation:** Obtaining suitable non-technical data is a challenge in most organisations. And while this may be a disadvantage, it is also a

strength in that SD modelling will point out an optimum set of measurements to make for good decisions.

**Unexpected behaviour:** When model behaviour differs from expectations it could point to model defects, or a new insight. This presents an opportunity for improvement.

***Drawbacks***

**Possible inconsistency:** Different groups of people with different backgrounds and experience levels could come up with widely different initial models of the same problem area. This is both a negative and a positive in that the different mental models of individuals could produce a more insightful outcome. Convergence to a common set of stocks and flows will occur after a few iterations.

**Overwhelming visual complexity:** Due to the ease of model extension, it could become difficult to comprehend large stock and flow networks. Tools such as iThink attempt to ease the task by allowing grouping of related stocks and flows, with the ability to blank out detail within a group.

**Obscuring of common concepts:** Sometimes in the conceptual phase, certain issues may arise which are not a direct or near fit with SD elements. In breaking down the concept to fit SD levels, flows and converters, the intent and coherence of the original concept may be lost. For example, 'temperament', is well understood in Human Resource settings, but in SD its multi-dimensionality will surface, and in broken-down form it would lose its impact and meaning.

**Data sourcing:** It becomes progressively more difficult to find precise data applicable to the lower, more minute levels, within a reasonable time. One

solution is to limit model component depth to what data can be easily obtained, and to construct hard-to-find data through the opinions and educated guesses of domain experts.

**Model iteration:** Model behaviour that is significantly different to expectations could point to model defects, or a new insight. Multiple iterations of model breakdown/rebuild and additional real-world data are then required to restore confidence in the model.

# Chapter 5 – How the TSO model was developed

## 5.1 SD model development process

Forrester (1991) proposes a model development process for an *existing system*, as described in Section 4.4. However, in this particular study, the purpose was to illustrate the applicability of SD simulation to technical support organisations. The process embarked on the development of an 'ideal' TSO structure by drawing on existing in-company data and mental models (experience and practice), and combining this with ILS 'best practice' principles, as per Fig 5.1 below. The resultant structure is encapsulated in an SD stock and flow model, using the iThink software tool.

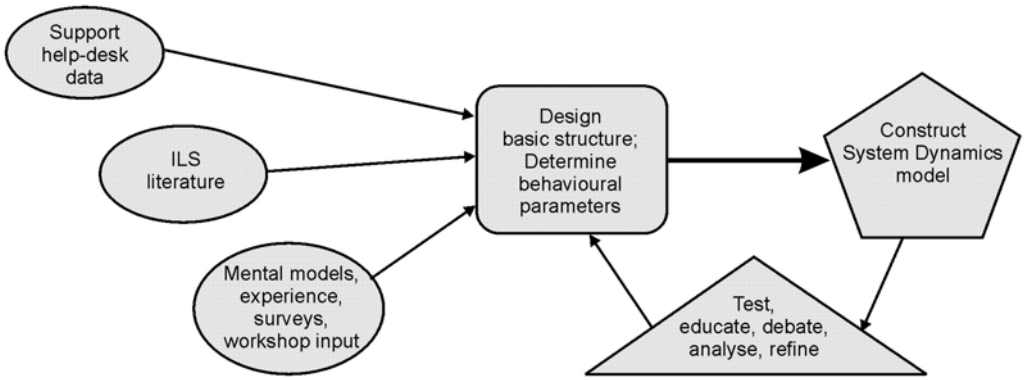



**Fig. 5.1**: TSO model development process

The integrity of the model is significantly increased through iteration, as each review/refinement cycle brings new understanding and insight.

Detailed steps are as follows:

**Step 1: Data gathering**

- Analyse the basic organisational structure to derive the significant building blocks of the group.
- Map out the work-flow and understand the influence of the organisational structure on workflow and effectiveness.
- Survey the key workflow support systems and their contribution.
- Collect and analyse at high level some sample workflow and productivity data.
- Determine key performance parameters and decide which should be the focus of the model.
- Mental models.

**Step 2: Develop the initial Stock and Flow model**

- Develop a first approximation model (best done with software tools such as 'iThink'.)
- Use the gathered data.
- Add 'mental models' or personal viewpoints of the problem as gained from experience or deeper study. (For example, in a service delivery context, what structures or behaviours or support functions would benefit a service organisation.)
- Build the model around one key operational parameter, for example the 'Problem Log'.
- Structure the input data to reflect general trends, and estimate data where no actual data is available.
- Normalise the data where possible, to resolve the difficulties that arise with absolute values.
- Document each item in some detail, to facilitate the reviewing process and model changes.
- Develop the model equations as determined by the modelling tool.

**Step 3: Run the model**

- Adjust input data and operating levels to produce a model behaviour that makes logical sense.
- Set up a convenient user interface, or 'control panel', to facilitate multiple runs, and for demonstration purposes.

**Step 4: Review the model**

- Consult with peers and domain experts for input on the model structure.
- Obtain rankings and weighting factors for components, to reflect their impact. For example, in the development of the TSO model, 'IT network availability' was initially regarded as very important. On surveying users and experts in the field, and combining this with actual uptime data, the network had an impact of 3%, compared to 15% for a 'technical documentation system'.

**Step 5: Refine the model**

- Adjust and refine the model.

*Go to Step 3.*

**Step 6: Implement the new policies or improved structure**

- This step is not covered in this study, but would be a natural next step if the proposed TSO structure was to be put into practice.

## 5.2 Drawing on the Support Logistics discipline

Support logistics is the discipline of building and managing organisations that support and maintain technical systems and equipment for industry. Also known as Integrated Logistics Support, the discipline arose out of the military, and although literature such as Jones (1987) refers to ILS primarily in the defence industry context, the concepts are applicable wherever sophisticated equipment and systems need to be kept in optimal running condition.

This research report proposes that the ideal TSO draw strongly from this tried and tested discipline, tapping into the wealth of experience and development that is available from ILS.

ILS specifies the following sub-disciplines – supported also by another key proponent, Blanchard (Blanchard, 1998) - as essential to a support organisation: Maintenance planning (would include service organisational planning); Manpower and personnel; Supply support; Support and test equipment; Training and training systems; Technical documentation; Computer resources; Packaging, handling, storage and transportability; Facilities, and Reliability and Maintainability analysis.

In the model proposed here, the components supply support, packaging, handling, storage and transportability, and facilities are not included in the TSO, as it is assumed to be supplied by the host organisation. Similarly, computer resources are considered ubiquitous, but 'network availability', a more challenging resource, is modelled instead.

### 5.3 Validity of the model

There is, as yet, no proven method to develop an SD model in a rigorous way to ensure validity. It is easily argued that if a few teams were given the same description of a situation to model, they would all develop very different models on the first pass. This may sound disconcerting at first, but is a strength in many ways. It points to the richness in people's mental models and emphasises the need for the educate/debate/refine step in the development process. It would be equally evident then that when the teams debate and review their versions with each other, a 'super model' could emerge, containing the wisdom, experience and real-world insights of the combined group, more likely than not more than the sum of the parts.

SD proponents argue that usefulness is more important than validity and that the relevance and value of a model is ultimately in the eye of the user (Shreckengost, 1984). Similarly, Forrester (1991) argues for adequacy for the purpose, and that essential areas for consideration are: appropriate level of detail (time and cost impact), setting of problem boundaries, and subjecting the model to in-boundary and out-of-boundary tests, to highlight any unexpected behaviours. Validity of a model is said to be achieved when it can be used with confidence.

### 5.4 Modelling style

The author's preferred modelling philosophy is biased towards symmetry and the exposure of important components to aid evaluation and discussion, at the cost of increased visual complexity. Other SD developers prefer less visual detail, by relegating factors to internal converter calculations; however, this approach could lead to some confusion and less clarity.

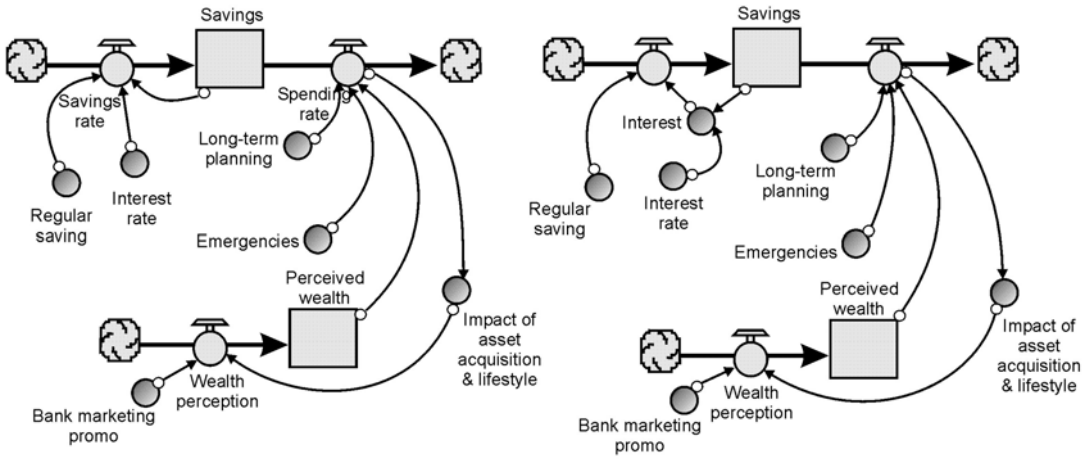



**Fig. 5.2**: Equivalent representations

Interest feedback loop details hidden in 'savings rate' equation.

**Preferred style:** An extra converter exposes the 'interest' component. 'Savings rate' becomes a simple sum.

# Chapter 6 – Full description of the TSO model

## 6.1 Introduction

An overview of the TSO model is given in Fig 6.1. The model is displayed in 'sector' mode, a feature of the 'iThink' tool that conglomerates low-level components to avoid confusion and complexity.

The general structure is that of a two-tier service and support organisation. The on-site, first-level maintenance and support team typically deals with 80% of everyday problems, of a less demanding nature. However, when it finds a difficult problem, it is escalated to the head-office support team, who, under normal circumstances, would have been responsible for the original installation and commissioning . They would also have direct access to the development team to help efficiently resolve difficult problems. Both teams use technical support infrastructures and systems aids that are also subject to availability and uptime. However, from experience, it is also expected that infrastructure support systems undergo continuous improvements over time, due to the critical impact on productivity of the human resources it serves.

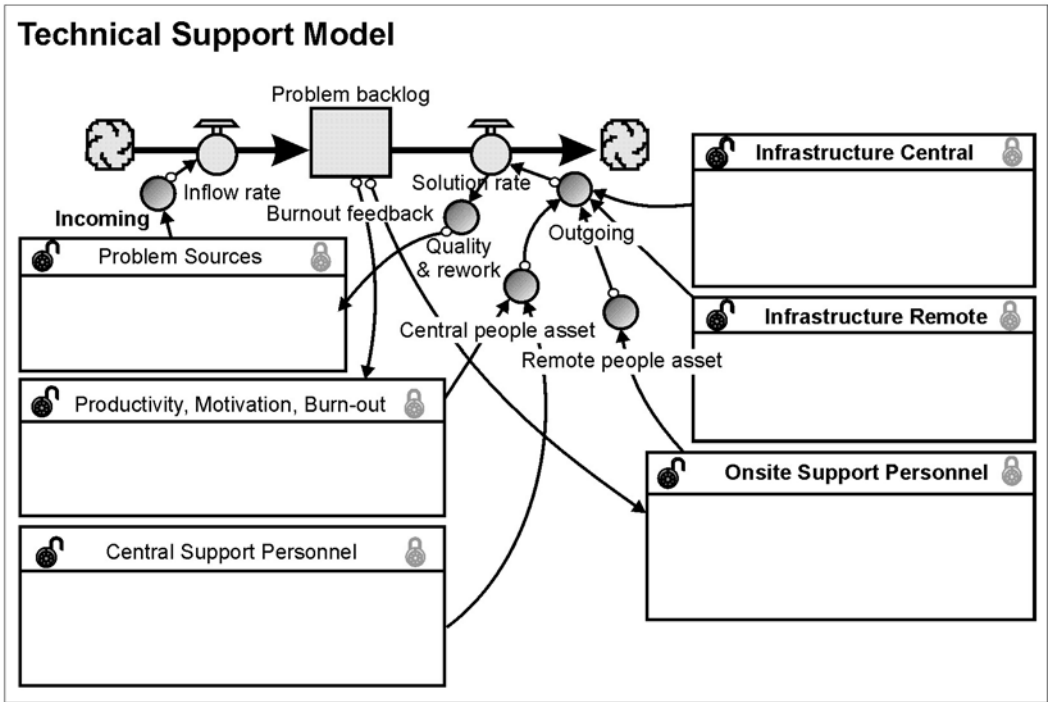



**Fig. 6.1**: TSO model overview

## 6.2 Sector overview

### 6.2.1 'Problem backlog'

Referring to Fig. 6.1, the central stock represents the highest-level entity being modelled. In a TSO setting, a well-managed backlog is a key performance area, and one which is most visible to TSO personnel and users alike. The aim is to deal with user problems and requests without panic or in a rush, but in a confident, controlled and unstressed manner. Associated model components are the inflow-rate and solution-rate 'valve' variables that collect the driving forces behind incoming problems and solved problems. Objects called 'converters' provide a generic method to combine outputs from sectors or other converters.

### 6.2.2 'Problem Sources'

This grouping represents the possible sources of problems, and is specific to the types of products and the environment being served by the TSO. The 'quality and rework' converter simulates the real world, feeding back a proportion of solved problems to the problem queue.

### 6.2.3 'Productivity, motivation and burnout' (PMB)

This sector models the dynamics of the human resource pool, the key actors in the resolution of problems. The current model assumes the dominant motivational impact to be the knowledge of the growth and decline of the problem-log, as shown by the 'burnout feedback' links. For emphasis, the PMB sector is accessible to the central team, but a similar structure is embedded in the 'Onsite Support Personnel' substructure.

### 6.2.4 'Infrastructure central/remote'

The 'Infrastructure' sectors contain the support systems that each of the central and remote groups need to assist them in their task.

### 6.2.5 'Central/Onsite support personnel'

These sectors contain the models for human resources supply, the pool of support specialists who are subject to ebb and flow due to acquisition and attrition. Typically, a training programme for new starters is part of the sub-system, with the rate of new entrants being driven by the exit rate, but with a large training time delay.
Sector internal details are shown in Section 6.5.

## 6.3 Data gathered for concept design

Data and information gathering is an essential task in model development, and is part of the 'concept development' phase, as per (Forrester, 1992). Table 6.1

below lists the data types, sources and other background information, which formed part of this study.

**Table 6.1:** Data collected for model development

| | Type of data | Description | Sources | Period | Data Quality | Model component relevance |
|---|---|---|---|---|---|---|
| 1 | General TSO issues. (Others' mental models) | A number of personal interviews and feedback workshops were conducted to gather information regarding the problems and opportunities in the product support arena, which were to be addressed by the proposed model of an 'ideal' TSO. | Colleagues. | - | Good. | General model structure. |
| 2 | Survey on proposed TSO support areas and weightings. | A survey was distributed listing the proposed components of a TSO model. Respondents were asked to add or delete any components as they saw fit, and to rank the importance of the various components. | In-company colleagues; other industry acquaintances, from Engineering and IT disciplines. | - | High. | Applicable to 'Support systems' submodel - structure and weightings (contribution) of each component. |
| 3 | Call desk data and trends. | In-company IT call-desk data was obtained and analysed for:<br>- volume of calls per period<br>- average resolution time<br>- peaks and troughs, relating to events and changes in the company.<br>- staff complement | In-company call-desk management system. | 1 year's call history. | Average. (Call-desk software requires an enormous amount of 'set-up' and transaction data entry, which few organisations seem to manage to do. The data required a significant amount of 'cleaning up', and was used only for | Calibration of problem call through-flow vs support personnel required. For general model insights only. |

| | Type of data | Description | Sources | Period | Data Quality | Model component relevance |
|---|---|---|---|---|---|---|
| | | | | | general trends.) | |
| 4 | Networking uptime. | WAN links from central to the outside world were considered by most specialists surveyed to be critical for the support of remote sites, but proved otherwise by the simulation*). The WAN provider's network uptime report provided some insight into uptimes being achieved. | External WAN service provider. (uptime report). | 1 year. | Good. | Support system weighting or impact of the 'network' component. |
| 5 | Support staff movements | Support personnel availability data was obtained, to gauge the stability or otherwise of the existing support team. | In-house product support department. | 1 year. | Average. Only gross data for the year was available, rather than statistics for daily/weekly availability support staff. | Human resource sub- model. |

*An example of Forrester's assertion that intuition, perception and mental models of individuals may be inaccurate. It needs to be verified by analysis and investigation, or ideally by simulation.

### 6.3.1 Simulated data

Although all the data required for model calibration and simulation runs were modified and 'shaped' to a degree, some elements of the model had no readily available data. In these situations, data had to be constructed from the author's experience and from informal input from colleagues and acquaintances.

The following types of data were simulated or estimated:

- New projects coming on-stream: A spate of new fault-calls would arise as new systems are installed.

- General and process changes: Ongoing operational changes to plant to overcome production bottlenecks could introduce a host of new problems.

- Machine ageing: As equipment wears out mean-time-between-failures (MTBF) decreases, causing a higher-level of fault-calls.

- Typical uptimes for: Knowledge base system, test rigs, fault diagnostic systems, call-desk, and documentation management systems.

- Typical logistical availabilities from internal or external organisations, such as spares supply, and external supplier support.

Where a component behaviour is hard to visualise, or subject to much variation, a random function is used. The impact of randomness is determined in the Sensitivity Analysis tests.

### 6.3.2 Visibility of simulation data

One of the major benefits of the SD model structure and the 'iThink' implementation is the visibility of input data. Fig. 6.2 below illustrates the iThink input data tool.

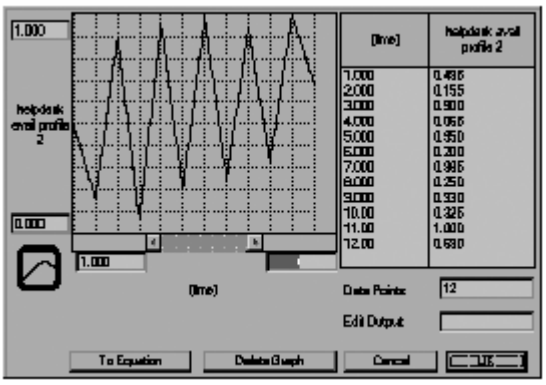



**Fig 6.2**: Example of an input data component: 'Call-desk availability'.

The graph shows the typical variation in Call-desk availability, but with a general upward trend or constant improvement of the service.

### 6.4 Implemented model structure

The iThink package provides a convenient three-layer modelling environment:

- A high-level user interface layer, which can be configured as a control panel, as per Fig. 7.1. This allows a view into the behaviour of the model as it runs, and the ability to manipulate parameters easily between reruns. Educational walkthroughs and user guides, with pop-up help texts

- The modelling layer, where the heart of the system - the stock and flow model elements – are implemented, Fig. 6.1, and,

- The equation layer, containing the equations underlying the stock and flow elements, given in Appendix D.

### 6.4.1 Documentation feature

Apart from the detailed configuration dialogues associated with each element in iThink, provision has been made for a descriptive text-box in which the model constructor can capture the thinking behind a particular element.

Example of documentation in the TSO model:

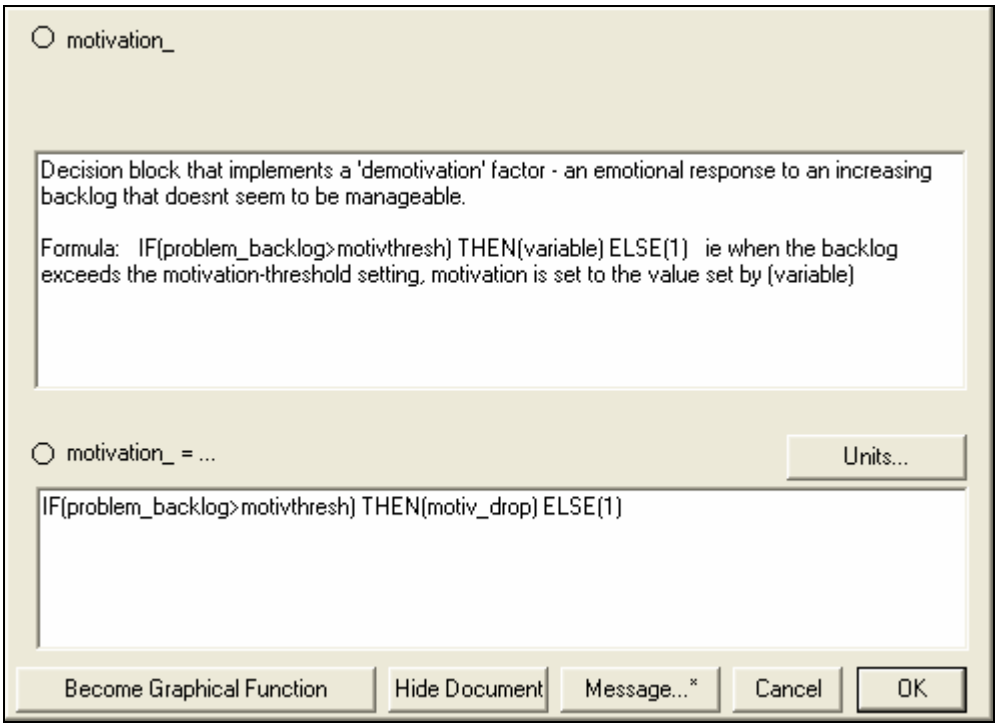



**Fig 6.3**: Documentation

### 6.4.2 Model equations

As described in Chapter 2, all SD models are mathematically described by integration and arithmetic equations. The 5th order model developed here contains about 115 equation lines. A full equation listing is provided in Appendix D.

### 6.4.3 Decision or logic blocks

Often in a real situation there is a need for non-linear constructs to cope with decision-type logic or threshold detection and triggering. In the TSO model, such a logic block is used to sense and set off the human behaviour ‘demotivation’.

Since motivation is an emotional response to what people hear and see, it rarely behaves linearly. A drop in motivation could be the result of a 'last straw' situation, for example. In the TSO model iThink's 'decision diamond' concept tries to emulate the central team's response when seeing the problem-log getting 'out of control' in their opinion.

*Decision equation*

motivation_ = IF(problem_backlog > motivthresh) THEN (motiv_drop) ELSE(1)

meaning that if the backlog exceeds a threshold (set at 500 nominally), then the 'motivation' constant in the PMB section drops to a value less than 1, and reduces the teams ability to solve problems at the normal rate.

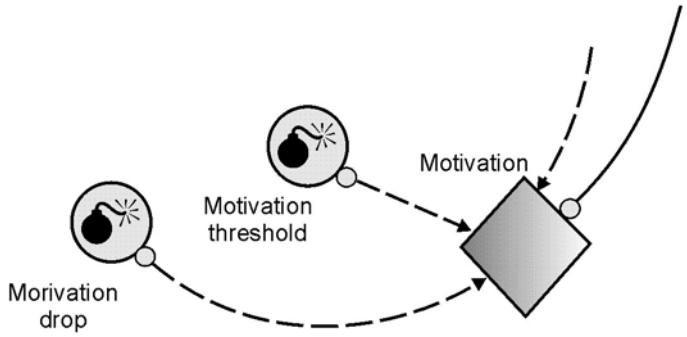



**Fig 6.4**: Decision logic in iThink

## 6.5 Detailed description of model sectors

The modelling tool iThink allows the complex-looking SD diagram to be split up into 'sectors', with each sector containing related factors or sub-models. Referring to the overview in Fig 6.5, the sectors are described as follows. The actual equations behind the diagrams are provided in Appendix D. Appendix C lists the graphical data and the reasoning behind it.

### 6.5.1 Sector: Main

The 'main' components are not in a sector. The left-hand side of the key 'problem backlog' stock contains all the factors that add to the problem log (inflow). The right hand side contains the factors that draw down on the problem log (outflow). The 'incoming' and 'outgoing' converters act simply as collectors for the contributing sectors.

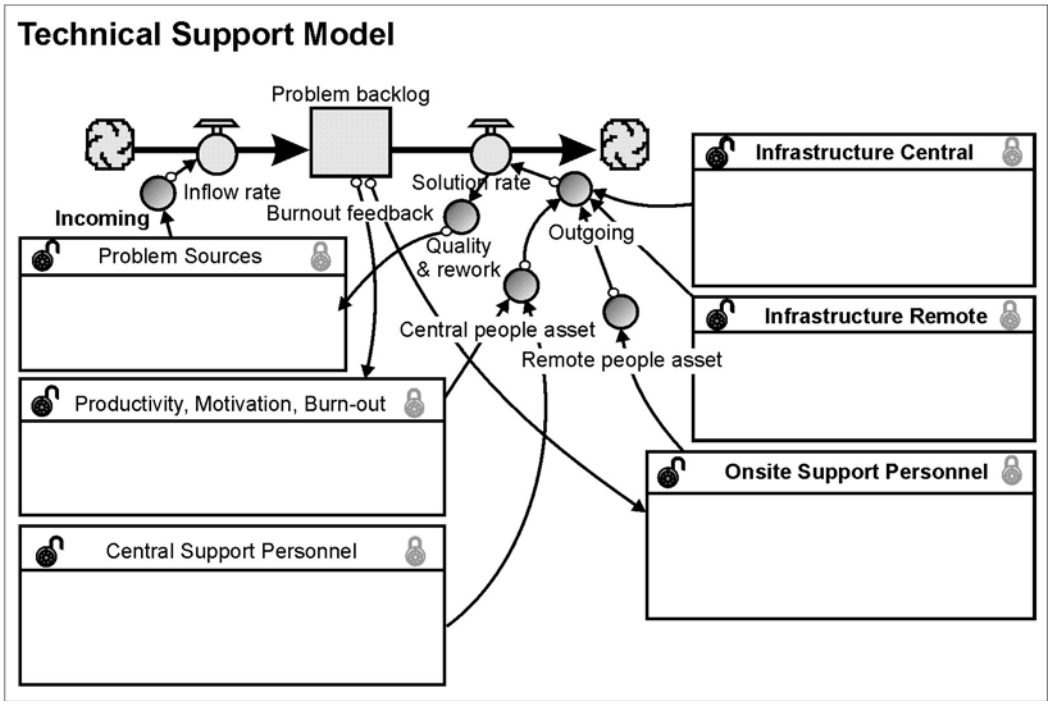



**Fig 6.5**: Model overview by sector

### 6.5.2 Sector: Problem sources

The 'problem sources' sources connector tie together causes of problems to be solved. Quality and rework deals with the feedback loop from the outflow rate to simulate the fact that a fraction of problems solved return to be dealt with again.

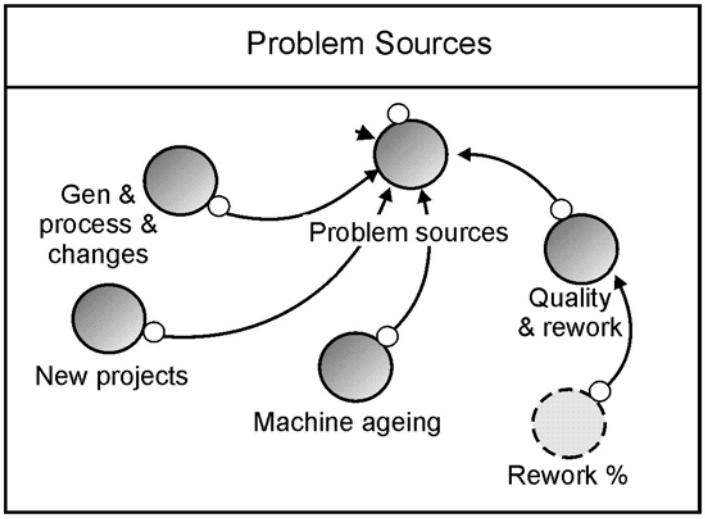



**Fig 6.6**: Problem sources

### 6.5.3 Sector: Productivity, Motivation, Burnout (PMB)

The PMB sector collects HR 'soft' issues that impact on the ability of the support people to operate at optimum productivity.

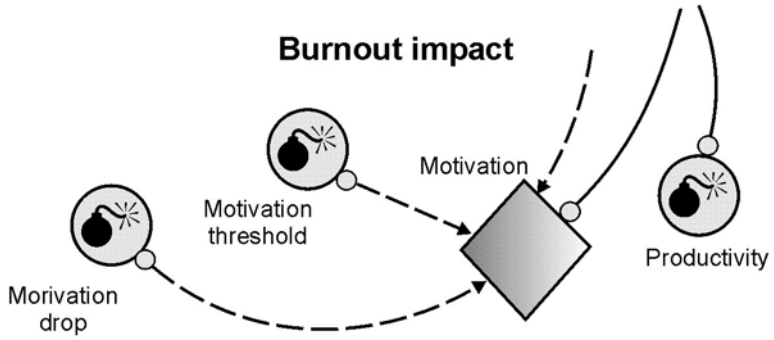



**Fig 6.7**: Productivity, motivation, burnout (PMB)

### 6.5.4 Sector: Central support personnel

The central support sub-model simulates typical team dynamics, people joining, training, becoming productive, and leaving. Many more dynamics could be added: leave, a random sick leave factor, etc; however, these contributions would typically be minimised by correct leave planning. The training programme is simulated with a delay stock – people enter, but become available only after a delay of 'programme duration'.

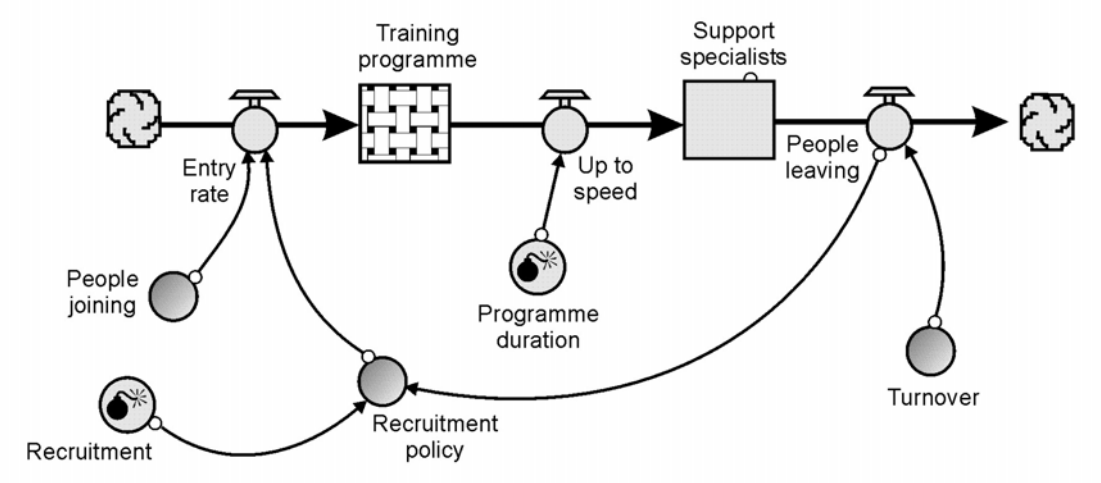



**Fig 6.8**: Central support personnel

### 6.5.5 Sector: Remote (or onsite) support personnel

The remote support people structure is identical to the central one, but includes a PMB component.

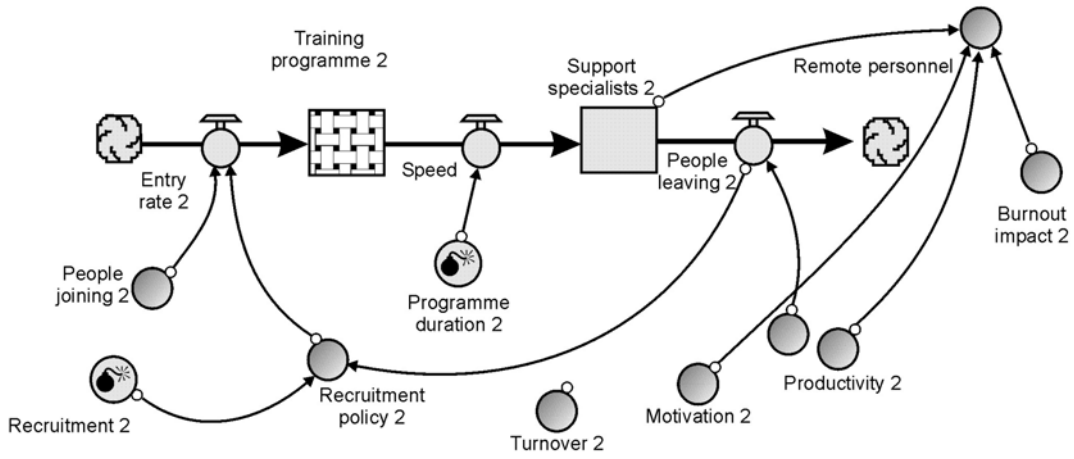



**Fig 6.9**: Remote support PMB

### 6.5.6 Sector: Central infrastructure

All the support systems contributors are contained in this model sector. The relative impact of the various facilities is described in Section 6.3.1. A general 'improvement' converter is included to cater for capital invested in upgrading

infrastructure. It also simulates the general improvement that arises as support people become more skilled in the use of their facilities.

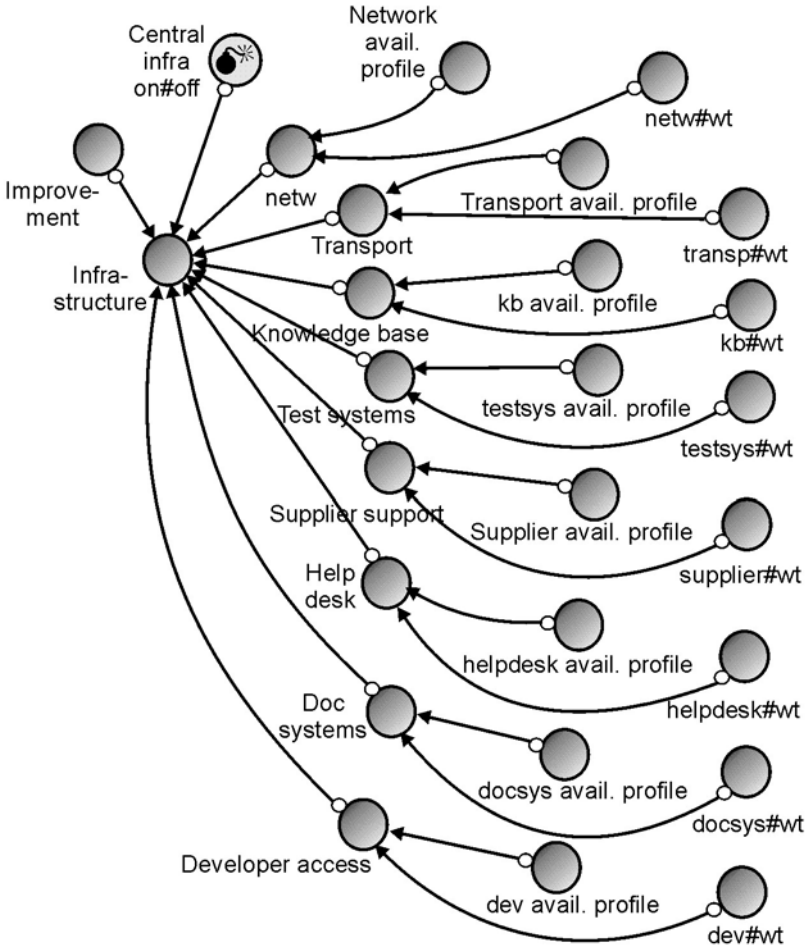



**Fig 6.10**: Central Infrastructure

### 6.5.7 Sector: Remote infrastructure

Again the remote system is similar to the central system, except for the absence of 'transport', 'network', 'developer access' and 'test systems'. More relevant facilities such as 'spares', 'built-in diagnostics', general 'fault diagnostic' systems and 'preventative maintenance' systems are included.

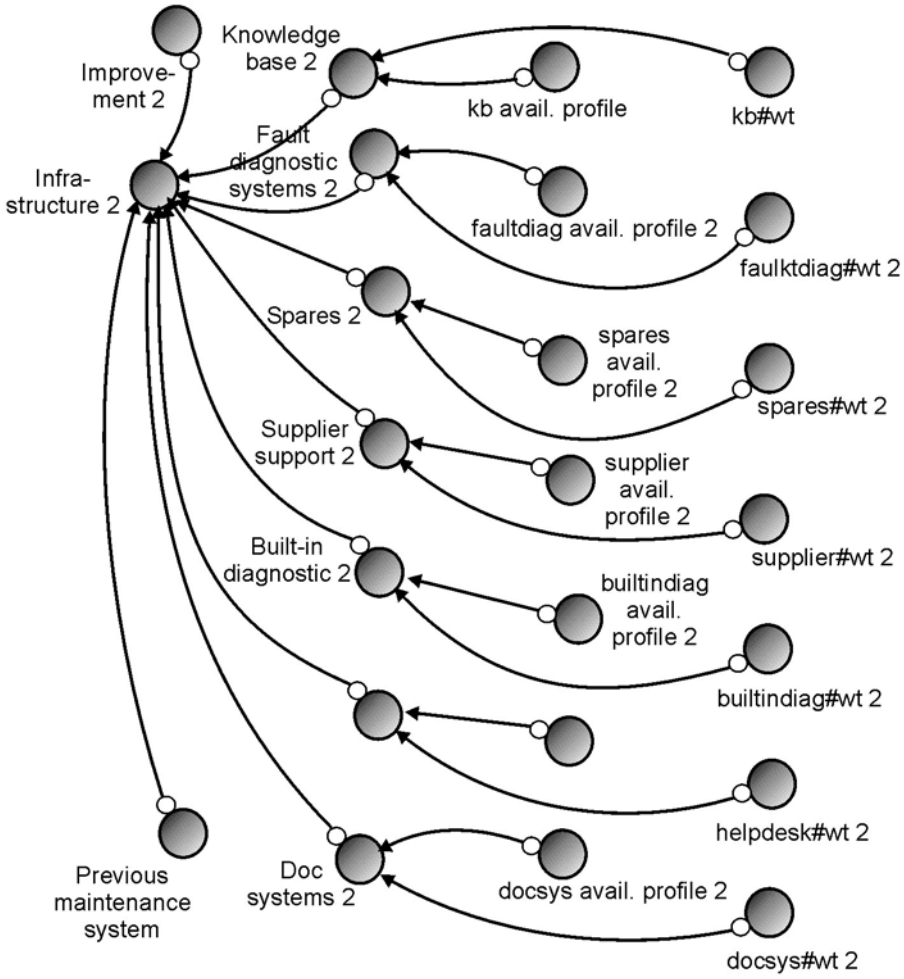



**Fig 6.11**: Remote Infrastructure

# Chapter 7 - Model testing and results

The following section describes the evaluation of the theoretical model.

## 7.1 User interface control panel

This essential feature provides the user with a straightforward view of key information about the model, much along the lines of a flight simulator 'dashboard'. iThink allows any value or variable to be monitored, and all constants (associated with connectors) to be manipulated.

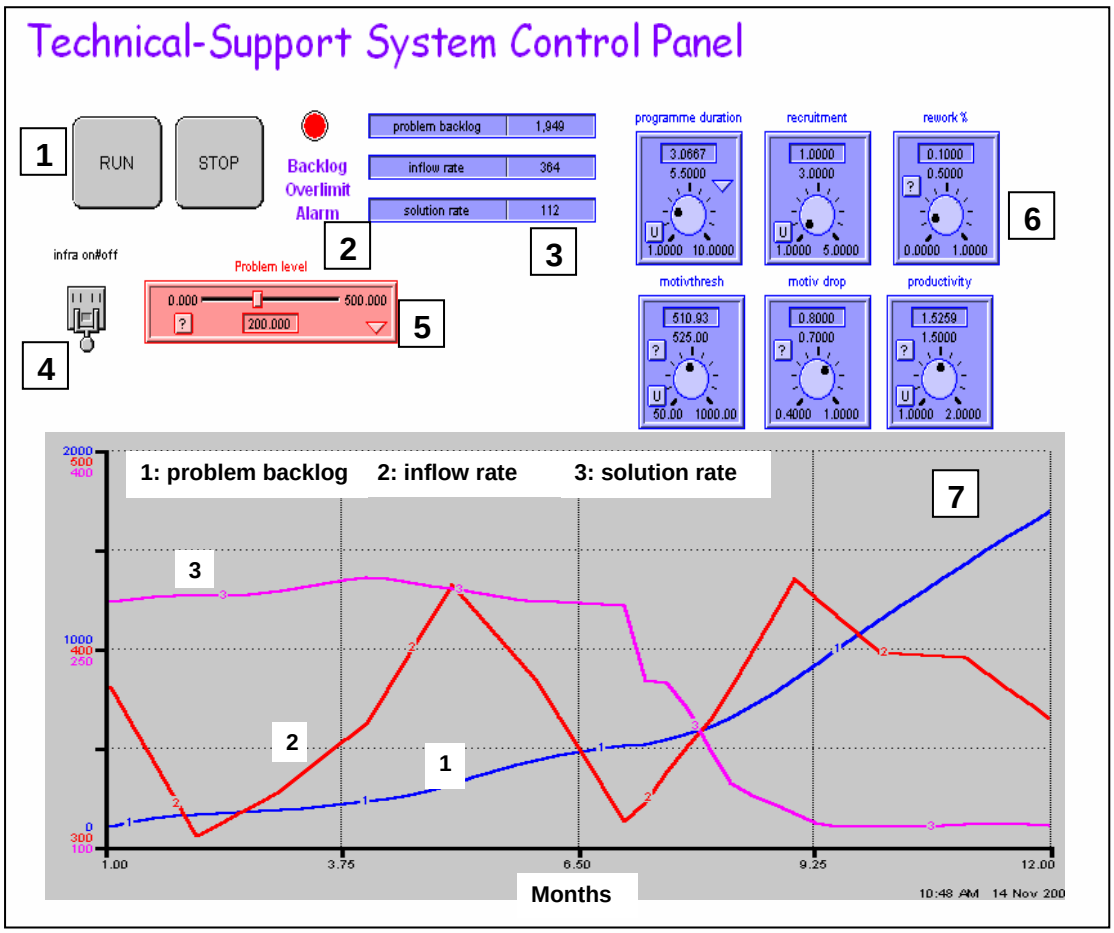



**Fig 7.1**: Simulation dashboard

Legend:

1. Run/stop buttons to control simulation runs. The simulation terminates after the equivalent of 12 months operating of the TSO (nominally 10 sec.).

2. Problem backlog alarm limit, attached to the main stock 'Problem Backlog'. The indicator is green under normal circumstances, turns yellow when backlog exceeds 500, and turns red and flashes when backlog exceeds 680. A pop-up warning will also appear at this point.

3. Numerical values for key variables.

4. On/off switch to simulate a major failure in the system. In this case, the switch has been attached to the 'Infrastructure' component (Fig. 6.10). Switching the toggle to 'off' will remove the contribution of the support systems at central from the problem-solving capability of the TSO, i.e. the right-hand side components to the problem stock.

5. Slider control for 'problem level', to set the number of problems per month the TSO is to deal with. This is useful for testing the ability of the system to deal with more than the nominal 200 problems, with the same personnel numbers and infrastructure contribution.

6. Similarly, 'volume control knob' devices are used for setting HR and training programme-related parameters. Unlike sliders, these cannot be adjusted during a simulation run.

7. A graph trending panel, for up to 5 variables.

## 7.2 Model validation

Four simple scenarios were set up to validate basic model behaviour. The outcomes can be assessed by intuition and a little background knowledge of the

service provision environment. Then follows a more extensive sensitivity analysis, where a wider range of parameters are manipulated to search for any non-intuitive behaviours, which could either mean (1), a modelling error (incorrect assumption, equation error etc.), or (2) a genuine non-obvious phenomenon that adds insight into the dynamics of a TSO.

### 7.2.1 Steady state

A steady-state condition is defined where the problem inflow rate is balanced by the outflow, i.e. the combination of people and support systems keeps the backlog under control. The simulation period chosen was to be 12 months, and backlog starting value was 100.

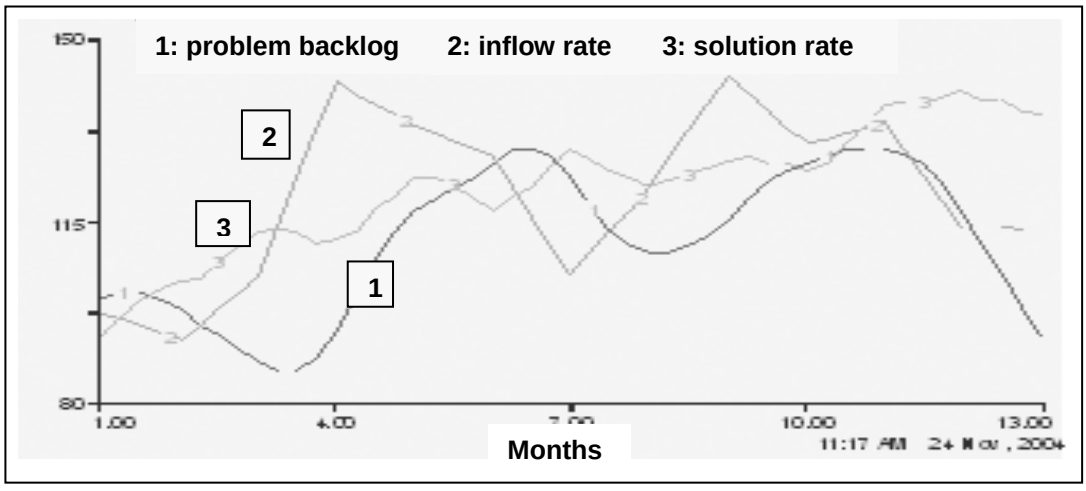



**Fig. 7.2**: Steady state conditions.

Trend 1, the problem backlog, initially at 100, varies ±25% and ends lower at about 90. This shows that over a period of one year the demand is well under control, possibly even with some spare capacity in the delivery system.
Trend 2, the problem inflow rate is the sum of the behaviours in the 'problem sources' sub-model. This trend shows a variation of ±20% around a mean of 120. The source of the variations can be found in the contributions of the problem-sources, many of which are set up as graphical input functions (Appendix C). For example, in month 4, the 'general and process changes' input was at its peak and 'new projects' was at the first of two peaks for the year, resulting in a sharp peak

in the stock inflow. The difference between inflow and outflow gives rise to a steep accumulation in the backlog at this point.

Trend 3, the solution rate, is the combined effect of people and support structures that solve problems as fast as they arrive. In this case, the trend is steadily increasing, indicating growth in problem-solving capability. The right-hand side of the model contributes to this behaviour, and by using the trend function further in iThink for the personnel sub-models, we find the central pool of support specialists increasing towards the end of the year due to trainees completing their programme, offsetting some losses in the remote personnel structure for the same time-period. For both the remote and central support systems, there is a general upward trend in contribution from most support components, except for the randomly variable components such as transport and spares availability, two areas with the least control. The combination of the 13 steady or increasing contributions from infrastructure results in an overall steady increase in resolution capability, with all other factors remaining equal.

*Rule of Thumb postulation*

*At this early stage, a general rule of thumb for control of a service operation backlog can be formulated: The outflow trend (resolution productivity) must cross the inflow trend (problem arrival) regularly. The crossings show a capability to keep up with demand, avoiding excessive accumulation that would normally lead to more rework and eventual runaway, requiring a large and costly intervention to correct.*

This insight is of the 'iceberg' variety since it could not have been reached without the underlying or hidden modelling work. It can be easily be seen that the alternative and traditional analysis methods – case studies, causal loops, and performance reports - would not have been able to easily show this phenomenon. Only a modelling concept with a mathematical basis that takes account of levels, rates and accumulation would be able to arrive at this conclusion.

### 7.2.2 Lowering the quality of work

Quality of repair work is adjustable via the 'rework%' slider. It is normally set at 10%, i.e. 1 in 10 problems will result in a 'come-back', a reasonable figure in high complexity environments, but a target for improvement. Fig 7.32 shows the behaviour of the system for a rework value of 20%, or 1 in 5 problems requiring rework. The backlog now shows a run-away situation, but interestingly only from month 7. The steady rise after that is the result of the integrative effect of the inflow being ahead of outflow for about 4 months (refer to rule of thumb above). The rework feedback loop correctly models the expectation that if quality of work deteriorates - possibly due to new entrants to the personnel pools - then more rework is required, adding to the inflow rate and forming a positive feedback loop. The obvious solution is to improve quality, which achieves two goals at once: the backlog is brought back within limits and service quality perception is improved.

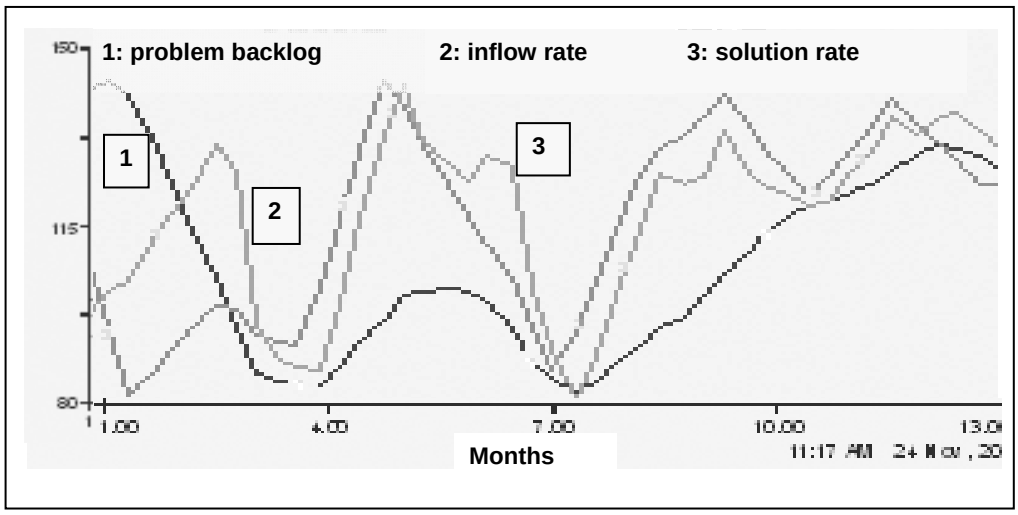



**Fig. 7.3**: Rework level at 20%

Another policy or solution that can be easily evaluated with the model in its current form (in line with Step 6 of the modelling process) is increasing the productivity levels for short periods (overtime etc), or introducing more staff (contractors), while service-quality issues are being resolved.

The SD model clearly shows the impacts and magnitude of such changes, which is not possible with the alternative methods.

### 7.2.3 Infrastructure failure

A gross failure of support infrastructure is simulated with an on/off switch attached to the central support systems, under steady state initial conditions. As shown in Fig. 7.4, the backlog runs out of control over a period of months rather than sooner, as problem-solving would draw on inherent knowledge and 'heroics'. However, without support systems, tasks take much longer and eventually there is a buildup that cannot be controlled without intervention.

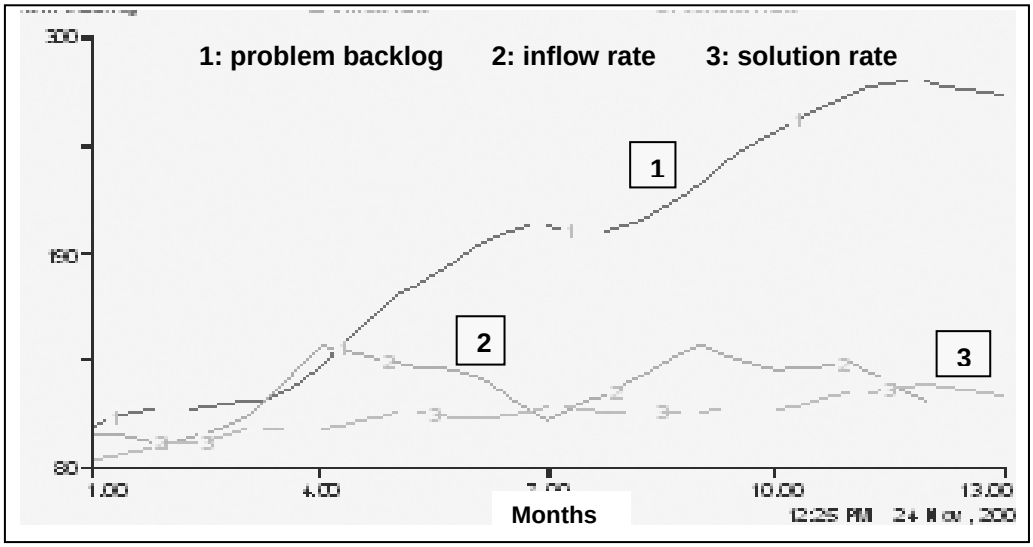



**Fig. 7.4**: Support systems failure

However, this is a worst case type failure, in practice individual subsystems only may fail.

### 7.2.4 Overload scenario and impact on Motivation

An overload scenario can be created with the 'problem-level' slider, which varies the problem throughput. Sustained overloading of people results in fatigue and general demotivation. The model includes a setting for motivation threshold which sets the level at which significant demotivation sets in. In addition, a 'motivation drop' control sets the level to which motivation actually drops (from 1

or no impact). A 'burnout' behaviour is also included and uses an X-Y function graph, as shown in Appendix C. Fig 7.5 shows the resolution capability of the central team diminishing when problem backlog reaches 150, or 50% above nominal. Inflow demand remains the same, but a runaway situation develops. For subsequent model iterations, these simplistic structures could be improved on, as suggested in Chapter 8.

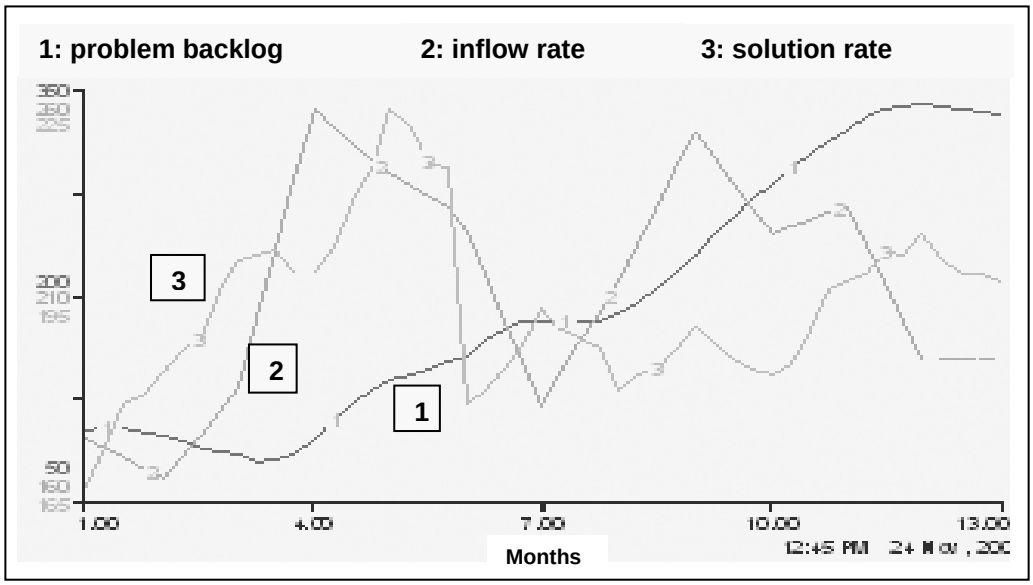



**Fig. 7.5**: Task overload

### 7.2.5 Sensitivity Analysis

Sensitivity analysis is a process used in simulation to test the response of the model to changes in variable parameters and constants. It would typically be used to test for redundant components or excess capacity, or when detrimental high sensitivity is encountered in order to reduce it. A number of adjustable parameters have been built into the model, some of which have been illustrated above. A more extensive analysis is presented in the table below.

**Table 7.1**: Sensitivity analysis: Varying other adjustable parameters

Legend:

1. THRU, Problem throughput
2. RW, Rework %
3. C.TP, Central training programme duration (months)
4. C.RCR, Central recruitment level (persons per year )
5. C.MTHR, Central motivation threshold (problem log level)
6. C.MDR, Central motivation drop (to fraction)
7. C.PROD, Central productivity level (0-1)
8. C.NETW, Central network availability (0-1)
9. R.PROD, Remote productivity level (0-1)
10. R.IMPR, Remote infrastructure improvement (0-1 per annum)

(g), graphical function

**Table 7.1**: Sensitivity analysis: Varying other adjustable parameters

| Case | | Parameters | | | | | | | | | | Problem backlog | Comment |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | |
| | | THRU | RW | C.TP | C.RCR | C.MTHR | C.MDR | C.PROD | C.NETW | R.PROD | R.IMPR | | |
| 1 | Base-case. | 100 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 92** | Under control, possible excess HR capacity. |
| 2 | Central infrastructure switched off. | 100 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 260** | Mild runaway situation, ave. of 13 added pm. |
| 3 | Rework level increase. | 100 | 30 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 385** | Warning alarm at 380, month 11. |
| 4 | Increase central training programme. | 100 | 10 | 5 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 137** | Mild increase, but with recovery. |
| 5 | Reduce central recruitment. | 100 | 10 | 3 | 1 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 140** | Mild, cyclical runaway trend. |
| 6 | Lower motivation threshold. | 300 | 10 | 3 | 2.5 | 200 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 518** | Motivation collapses to 0.7 at month 11. |
| 7 | Central motivation drop on overload. | 400 | 10 | 3 | 2.5 | 400 | 0.5 | 1 | (g) | 1 | 0.1 | **100 to 535** | Motivation collapse at month 11. |
| 8 | Productivity increase with case 3. | 100 | 30 | 3 | 2.5 | 400 | 0.7 | 1.3 | (g) | 1 | 0.1 | **100 to 248** | Significant recovery at end of period. |
| 9 | Central network off. | 100 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | 0 | 1 | 0.1 | **100 to 104** | Minimal effect. |
| 10 | Remote productivity down (e.g. due to strike). | 100 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 0.6 | 0.1 | **100 to 216** | Mild runaway situation, ave. 9 added pm. |
| 11 | Remote infra., no improvement, 10% to 0%. | 100 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0 | **100 to 113** | Minimal impact. |
| 12 | Problem overload 1. | 400 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 487** | Motivation collapses to 0.7 at month 11. |
| 13 | Problem overload 2. | 500 | 10 | 3 | 2.5 | 400 | 0.7 | 1 | (g) | 1 | 0.1 | **100 to 953** | Runaway. Motivation collapses to 0.7 at month 9. |

In all cases, the model behaved as expected in terms of general behaviour, and the model provided the exact level of impact, not predictable by intuition or inspection.

# Chapter 8 – Conclusions and further work

Intuitively and quantitatively, the model behaved as expected. Qualitative validation was performed through in-company reviews by support and maintenance logistics specialists, who corroborated the model and its approach. The author is confident that the model, even in its current 'first approximation' form, can be used in a live Technical Support Organisation setting to predict general performance trends and provide a high-fidelity alternative to current methods.

## 8.1 Suggestions for refinements and further work

It is expected that, when placed in a real-world situation, many refinements and extensions will be added over time, as users gain a greater understanding of the TSO business processes and their inherent complexity and connectedness. This is consistent with the 35 years of development of the SD field (Forrester, 1991) – SD modelling involves an iterative process.

The following refinements are mooted as logical next steps:

- Currently, a simplistic single backlog stock is used for all problems. A split in terms of problem priority, difficulty level and technology area is more realistic.

- Similarly, the resource pools should be diversified into separate stocks to distinguish between the various skills, areas of speciality and experience of personnel.

- Personnel burnout models can be improved by taking into account the time-periods for which overloads or higher demands are in effect.

- Costing dynamics should be added, especially for longer-term simulation, to deal with multi-year budget cycles and business economic cycles.

- Training programs are seldom of the 'oven' type, where trainees enter the programme and emerge fully trained and capable. The model should make trainees available to deal with certain classes and priorities of problems as part of in-service training.

- As solved problems are committed to the 'Knowledge Base' history, it improves problem-solving capability. However, knowledge 'setbacks' can occur, caused by equipment and product improvements and new models, which invalidate portions of the database. This behaviour is to be included by linking 'Product improvement' contributions back to the Knowledge Base converter, with a negative bias.

- There should be a clearer demarcation between Corrective Maintenance and Preventative Maintenance activities and the impact of each. The current model deals mainly with corrective aspects. Preventative Maintenance improvements have a high likelihood of reducing certain types of input problems.

- A reporting facility showing historical and projected trend-graphs should be added to the user interface to facilitate management communication with their teams.

## References

Blanchard, Benjamin S. (1998). “Logistics Engineering and Management”, Fifth edition, Prentice Hall, Upper Saddle River, NJ, USA, pp. 5-10.

Foo, Schubert, Siu Cheung Hui, Peng Chor Leong and Shigong Liu (1999). “An integrated help desk support for customer services over the World Wide Web — a case study” (First author affiliation: School of Applied Science, Nanyang Technological University, Singapore). Computers in Industry , vol. 41, issue 2, pp. 129-145.

Forrester, Jay W. (1991). “System Dynamics and the Lessons of 35 Years”, paper, D-4224-4, Sloan School of Management, M.I.T., pp. 23-25.

Forrester, Jay W. (1992). “System Dynamics, Systems Thinking, and Soft OR”, paper, D-4405-1, Sloan School of Management, M.I.T., pp. 3-6.

Jones, James V. (1987). “Integrated Logistics Support Handbook”, TAB Books, Blue Ridge Summit, PA, USA, pp. 1-4.

Kondratieff, Nikolai D. (1984). “The Long Wave Cycle”, Richardson and Snyder, New York, NY, USA.

Oliva, Rogelio & Sterman, John D. (2001). “Cutting Corners and Working Overtime: Quality Erosion in the Service Industry” (First author affiliation:Harvard Business School, Boston, MA, USA). Management Science Vol. 47, no. 7, pp. 894-914.

Shreckengost, Raymond C. (1984). “Dynamic Simulation models: How valid are they?”, ref. D-4463, National Institute on Drug Abuse, Rockville, Maryland, USA, pp 1.

Senge, Peter M. 1990. “The Fifth Discipline”, Doubleday, New York, NY, USA.

iThink/Stella technical documentation CD, v 7.0.3.

HEAT helpdesk system, http://www.frontrange.co.uk/ProductsSolutions/SubCategory.aspx?id=35&ccid=6, 19 March, 2006.

Footprints helpesk system, http://www.unipress.com/footprints/, 19 March, 2006.

## Bibliography

http://www.systemdynamics.org/short_bibliography.htm, 19 March, 2006.

# APPENDICES

**Appendix A – Glossary of terms**

**Appendix B – The 'iThink' software modelling tool.**

**Appendix C – Model data**

**Appendix D – Model equations**

## Appendix A – Glossary of terms

| | |
|---|---|
| Call-desk | A system used in after-sales customer servicing to receive and log problem queries, and to assign resources to resolve the problem. Almost exclusively implemented in software, with extensive statistics collected. |
| Causal Loops | The diagrammatic tool of the 'Systems thinking' approach to systems analysis. |
| Help-desk | See call-desk. |
| ILS | Integrated Logistics Support, the generalised modern-day maintenance and supply logistics discipline. |
| IT, IS | Information Technology, Information Systems. |
| iThink | An SD modelling and simulation tool, similar to associate product 'STELLA', produced by ISEE systems, formerly Hi-Performance Systems (HPS). |
| MTBF | Mean time before failure. |
| SD | System Dynamics, a modelling and simulation methodology application to complex systems. |
| Systems Thinking | A systems descriptive technique using Causal Loops diagrams. |
| TSO | Technical service or support organisation. |
| Uptime | Availability for service of equipment. |
| WAN | Wide Area network. |

## Appendix B – The 'iThink' software modelling tool.

### Page 1

# iThink™

## Systems Thinking for Business

*"Systems Thinking software like iThink is an increasingly valuable tool for constructing understanding about all kinds of dynamic systems from natural environments to team dynamics to economic markets."*

— Peter Senge, Author of The Fifth Discipline, Founding Chairperson, Society for Organizational Learning

Businesses must change to survive and grow. Why?

- Once-productive processes bow to new competition
- New markets challenge current methods
- Government regulation compliance adds work and cost
- Labor market shifts inspire new employee recruitment and retention programs

Change implies risk. What if the change doesn't work? What if things get worse? If the pilot project fails, what do we try next?

*"It is estimated that more than 75% of reengineering efforts do not produce targeted performance efforts."*

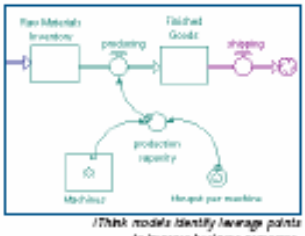
— Barry Richmond, Founder of isee systems

### Improving Business with *iThink*

*iThink* offers a risk-free way to make decisions that drive business improvement. Using *iThink*, you and your business team will create models that simulate business processes and scenarios; pointing out the impacts of a new procedure or policy, and offering opportunity to fix undesirable outcomes. Based on Systems Thinking, *iThink* models serve as "practice fields," showing you outcomes that could be painful, costly, or embarrassing if discovered in reality.

### See the Big Picture

What if you radically increased sales and marketing efforts without adding network bandwidth? Website hits and download demands would go up (good), networks would go down (bad). *iThink* helps you identify key leverage points for improving business performance without losing sight of unintended consequences.




*iThink models identify leverage points to improve business processes.*

Going beyond spreadsheets and other linear approaches to business planning, *iThink* provides a whole system or big picture view of your entire operations. These views are based on dynamic modeling and Systems Thinking; all points of a system are included and can be examined or changed in any order. You avoid making decisions that have negative impacts on seemingly unrelated processes.

### Create Shared Insight

*iThink* is a powerful tool for communicating interdependencies between processes and problems. Your entire business team will understand the variables that impact your business. Shared insight enables teams to work together, further ensuring that decisions are fully implemented and mitigating risk.

### Customer List

These are just a few of the organizations using iThink models to improve business performance.

Accenture
Americredit Corp
A.T. Kearney
Bayer SpA
Boston Consulting Group
Bristol-Myers Squibb
British Telecom
Canadian Ministry of Health
Dow Chemical
DuPont
Eastman Kodak
Eli Lilly
Equity Residential
Ford Motor Company
General Dynamics
General Motors
Hewlett Packard
Honeywell
Kaiser Permanente
Kennedy Space Center
Keynote Financial Services
London Business School
McKinsey & Company
MetLife
Pfizer
Raytheon
Rohm and Haas
Samba Financial Group
Sanofi Aventis
Seagate Technology
Shell
Shering Plough
Sony Pictures
Telcordia
Towers Perrin
US Air Force
US Coast Guard
Verizon
Warner Lambert
Wyeth-Ayerst Pharmaceuticals

**Page 2**

Use *iThink* to:

- Reduce the risk of policy or process change
- Identify key leverage points for improving business performance
- Build models that simulate your business
- Create what-if scenarios for decision support
- Develop shared understanding across functional teams

*"We estimate iThink models helped the International Olympics Committee save about $18 million on the Athens Olympic Games."*

— Peri Loucopoulos, Professor, The University of Manchester

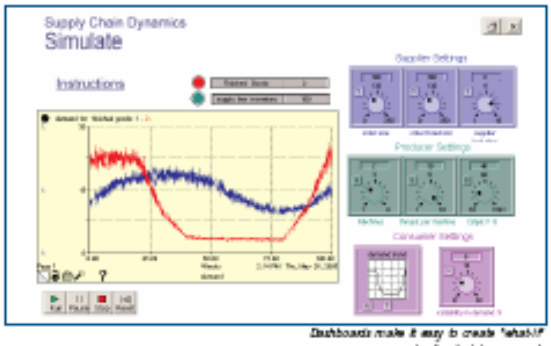
## Key Features

### Mapping and Modeling

- Stock and Flow diagrams support the common language of Systems Thinking and provide insight into modeled business processes
- Enhanced stock types enable discrete and continuous process with support for queues, ovens, and enhanced conveyors
- Model equations are automatically generated and made accessible beneath the model layer
- Built-in functions facilitate mathematical, statistical, and logical operations
- Arrays simply represent repeated model structure
- Sub-models support hierarchical model structures

### Simulation and Analysis

- Simulations "run" processes represented in models
- Sensitivity analysis reveals key leverage points and optimal conditions
- Partial model simulations focus analysis on specific sectors of the model
- Results presented as graphs, tables, animations, QuickTime movies, and files




*Dashboards make it easy to create "what-if" scenarios for decision support.*

### Communication

- Flight simulators and dashboards describe model components and facilitate manipulation
- Input devices include knobs, sliders, switches, and buttons
- Output devices highlight outcomes with warning flashers, text, graphs, tables, and reports
- Storytelling supports step-by-step model unveiling
- Sketchable graphs allow easy comparison of expected results with actual simulations
- Save as Runtime option creates full-screen, runtime models
- Multimedia support triggers graphics, movies, sounds, and text messages based on model conditions
- Model security features allow locking or password protection

### System Requirements

**Windows PCs Recommended**

233 MHz Pentium
Windows 98/2000/XP (English Version)
English version only
64 MB RAM
70 MB hard disk space
16-bit color
Soundblaster-compatible sound card
QuickTime 4 or greater

**Macintosh Recommended**

120 MHz PowerPC
Mac OS 9.2.2 or higher (English Version)
128 MB RAM
70 MB hard disk space
Thousands of colors
Quicktime 4 or greater

isee systems

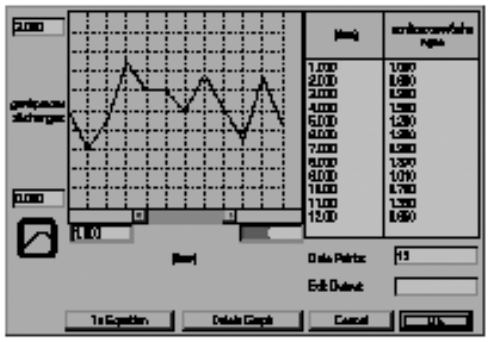
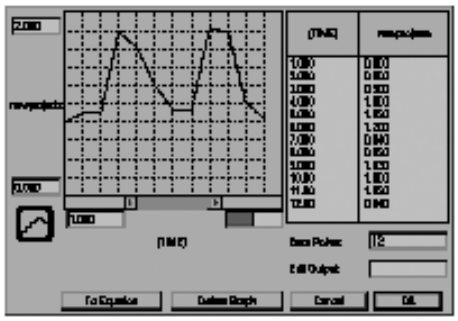
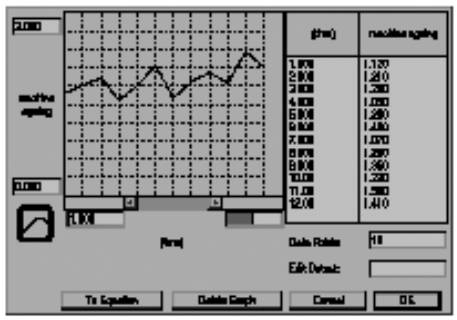
Phone 603 449 0090 Fax 603 449 0092
www.iseesystems.com

isee systems (formerly High Performance Systems) is the world leader in Systems Thinking software. Founded in 1985, isee developed STELLA, the first software to bring Systems Thinking to the desktop. In addition to STELLA, which is used primarily by educators and researchers, isee offers *iThink* for business simulation. isee is a privately-held company with substantial global reach in business, education, and government markets.

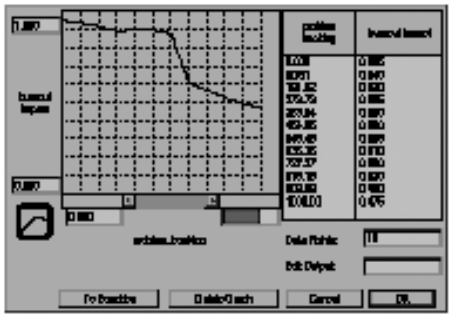
## Appendix C – Model data

Attached is an inventory of input and behavioural data as constructed for the model.

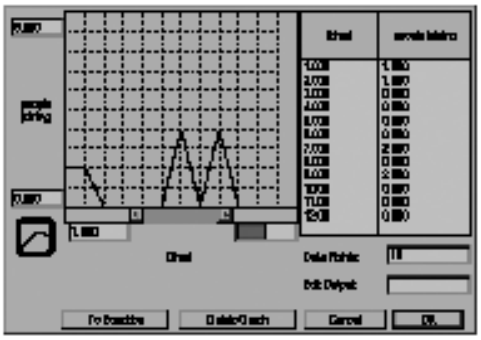
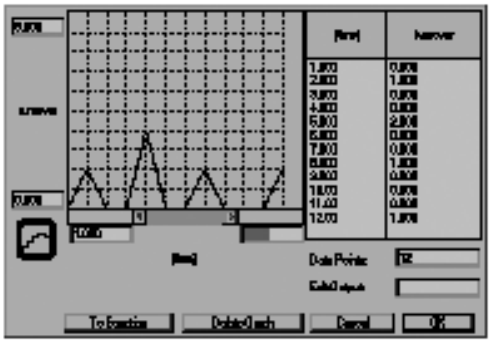
### C.1 Problem inflow components

| | |
|---|---|
| 1. General and process changes<br>These are part of 'problem generating' behaviours. In a typical large mine or manufacturing concern, many changes are made to the operation on a monthly basis. Examples: safety improvements; production de-bottlenecking, new equipment installations, system upgrades, etc. Most system level changes will cause the arrival of more calls for assistance, due to the knock-on effect of interconnected, complex sub-systems. A random function would also be a realistic indication of this source |  |
| 2. New projects<br>Typically a mining site may have two major projects a year and the graph gives an indication of the additional calls that may result. |  |
| 3. Machine ageing<br>Failures due to machine wear and tear become more random with an increase in quantities. However it is reasonable to postulate a general increase in problem calls due to this factor over time. |  |

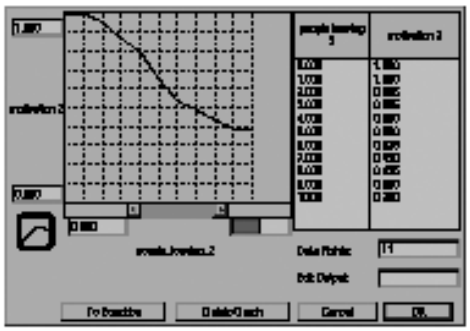
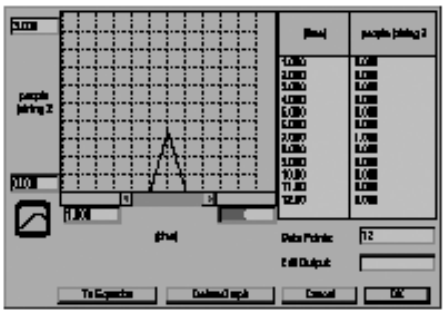
### C.2 Productivity, Motivation, Burnout (PMB)

| | |
|---|---|
| 4. Burnout impact<br>Burnout is modelled as a reducing factor, with a minimum impact of 1, and maximum of 0. Burnout increases gradually until a backlog of 500, at which there is a rapid drop. |  |

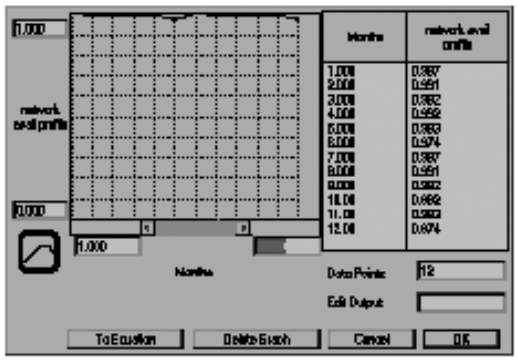
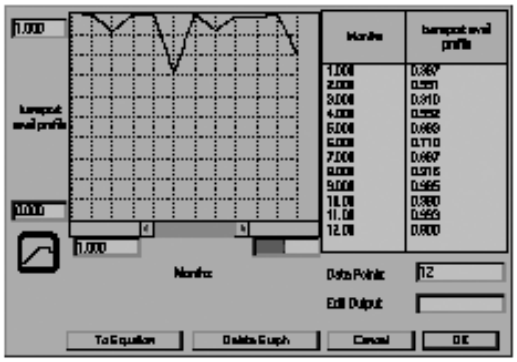
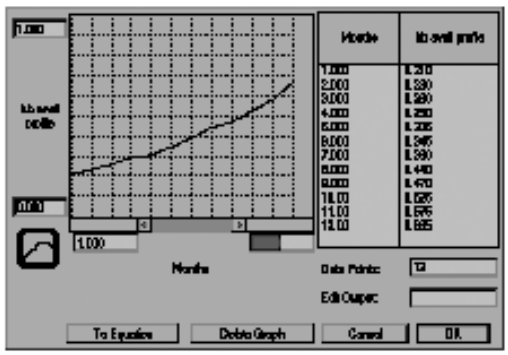
### C.3 Central support

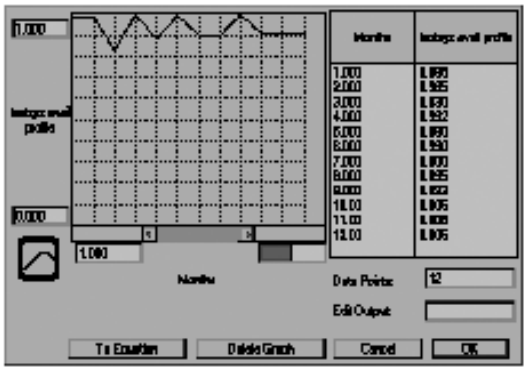
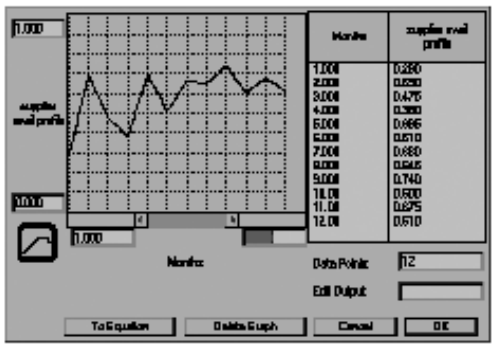
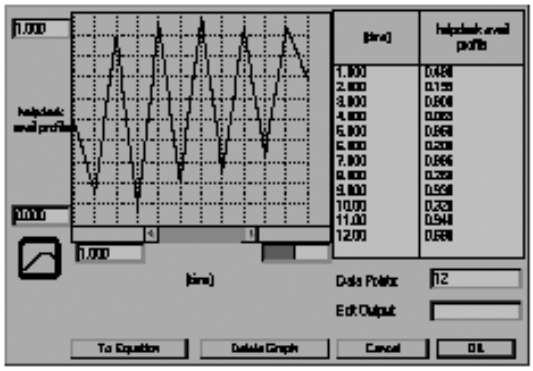
| | |
|---|---|
| 5. People joining<br>The graph shows a total number of 6 people joining in the year. |  |
| 6. People leaving (turnover)<br>The graph shows a total number of 5 people leaving in the year. |  |

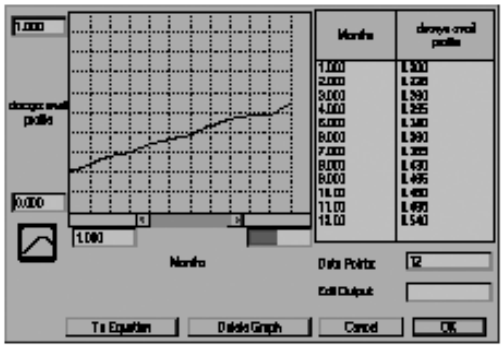
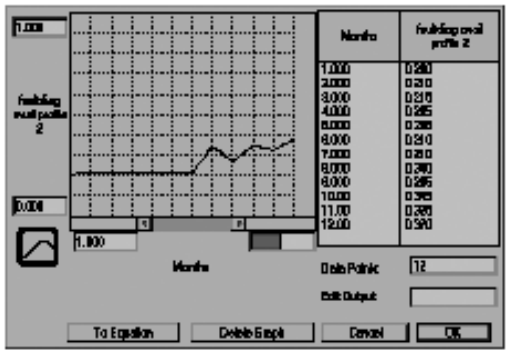
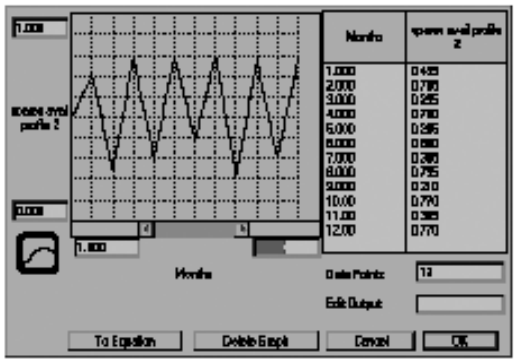
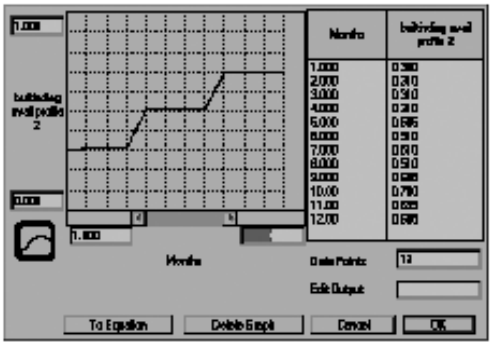
### C.4 Remote personnel

| | |
|---|---|
| 7. Motivation<br>Due to shortages of staff on remote mines, overloading is common, and motivation is impacted severely by people leaving. |  |
| 8. People joining<br>Only 1 person joined in the year, a typical scenario in remote mines which struggle to fill positions. |  |

### C.5 Infrastructure components

| | |
|---|---|
| 9. Network availability<br>(Real data) Data network availability is fixed within boundaries through a contract with the service provider (Telkom). |  |
| 10. Transport availability<br>Transport is generally available, but with possible serious dips and delays, especially to remote sites. |  |
| 11. Knowledgebase availability<br>It is expected that an expanding knowledgebase will positively impact the ability to solve problems. |  |

| | |
|---|---|
| 12. Test system availability<br>Test systems are shared and queues may form. Technical unavailability is caused by modifications, upgrades, etc. |  |
| 13. Supplier availability<br>Arbitrary profile, but overall supplier availability can vary substantially, albeit with an expected general improvement over time due to customer pressure. |  |
| 14. Help-desk availability<br>Arbitrary profile, but help-desk efficiency can vary widely, but again with a high probability of improvement over time due to user demands. |  |

| | |
|---|---|
| 15. Documentation system<br>A continuous improvement and impact is expected over time. |  |
| 16. Fault-diagnostic system<br>A continuous improvement and impact is expected over time. |  |
| 17. Spares availability<br>Possible random behaviour, as stores systems are single source, with possibly many demands on common items. |  |
| 18. Built-in diagnostics<br>Built-in diagnostics makes fault-finding easier, and serve as a training tool. Fault-finding efficiency is expected to improve over time. |  |

## Appendix D – Model equations

*(Note: Construct types are described on first occurrence only.)*

| | Equation | Description (first occurrence) |
|---|---|---|
| | ***Sector: Central Support Personnel*** | |
| 1. | support_specialists(t) = support_specialists(t - dt) + (up_to_speed - people_leaving) * dt | Stock integrator for personnel. |
| 2. | INIT support_specialists = 10 | Initialisation. |
| 3. | up_to_speed = CONVEYOR OUTFLOW | Special stock type 'conveyor' for training programme. |
| 4. | o TRANSIT TIME = programme_duration | Transit-time or speed of conveyor = training programme duration. |
| 5. | people_leaving = turnover | Equivalence formula required for visual simplicity. |
| 6. | training_programme(t) = training_programme(t - dt) + (entry_rate - up_to_speed) * dt | Conveyor integrator for training programme. |
| 7. | INIT training_programme = 3 | Initialisation. |
| 8. | o TRANSIT TIME = varies | Slider attached to conveyor. |
| 9. | o INFLOW LIMIT = INF | No limit. |
| 10. | o CAPACITY = INF | No limit. |
| 11. | entry_rate = people_joining*recruitment_policy | Formula. |
| 12. | up_to_speed = CONVEYOR OUTFLOW | |
| 13. | o TRANSIT TIME = programme_duration | |
| 14. | programme_duration = 3 | Constant. |
| 15. | recruitment = 2 | |
| 16. | recruitment_policy = people_leaving*recruitment | |
| 17. | people_joining = GRAPH((time)) | **Graph of quantity** over time, and **graph data.** |
| 18. | (1.00, 1.00), (2.00, 1.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.00), (6.00, 0.00), (7.00, 2.00), (8.00, 0.00), (9.00, 2.00), (10.0, 0.00), (11.0, 0.00), (12.0, 0.00) | |
| 19. | turnover = GRAPH((time)) | |
| 20. | (1.00, 0.00), (2.00, 1.00), (3.00, 0.00), (4.00, 0.00), (5.00, 2.00), (6.00, 0.00), (7.00, 0.00), (8.00, 1.00), (9.00, 0.00), (10.0, 0.00), (11.0, 0.00), (12.0, 1.00) | |
| 21. | | |
| 22. | ***Sector: Infrastructure-Central*** | |
| 23. | centralinfra_on#off = 0 | Boolean (binary) constant. (on/off switch) |
| 24. | dev#wt_ = .061 | |
| 25. | developer_access = dev_avail_profile*dev#wt_ | |
| 26. | dev_avail_profile = random(0,1) | Random function, producing value between 0 and 1. |
| 27. | docsys#wt = .167 | |

| | Equation | Description (first occurrence) |
|---|---|---|
| 28. | doc_systems = docsys_avail_profile*docsys#wt | |
| 29. | helpdesk = helpdesk_avail_profile*helpdesk#wt | |
| 30. | helpdesk#wt = .152 | |
| 31. | improvement = 1.1 | |
| 32. | infrastructure = improvement*((transport+developer_access+netw+doc_systems+helpdesk+knowledge_base+supplier_support+test_systems))*centralinfra_on#off | |
| 33. | kb#wt = .045 | |
| 34. | knowledge_base = kb_avail_profile*kb#wt | |
| 35. | netw = network_avail_profile*netw#wt | |
| 36. | netw#wt = .03 | |
| 37. | supplier#wt = .121 | |
| 38. | supplier_support = supplier_avail_profile*supplier#wt | |
| 39. | testsys#wt = .076 | |
| 40. | test_systems = testsys_avail_profile*testsys#wt | |
| 41. | transp#wt = .091 | |
| 42. | transport = transport_avail_profile*transp#wt | |
| 43. | docsys_avail_profile = GRAPH(time) | |
| 44. | (1.00, 0.2), (2.00, 0.235), (3.00, 0.28), (4.00, 0.295), (5.00, 0.34), (6.00, 0.36), (7.00, 0.385), (8.00, 0.43), (9.00, 0.465), (10.0, 0.48), (11.0, 0.485), (12.0, 0.54) | |
| 45. | helpdesk_avail_profile = GRAPH((time)) | |
| 46. | (1.00, 0.495), (2.00, 0.155), (3.00, 0.9), (4.00, 0.065), (5.00, 0.95), (6.00, 0.2), (7.00, 0.985), (8.00, 0.25), (9.00, 0.93), (10.0, 0.325), (11.0, 0.94), (12.0, 0.69) | |
| 47. | kb_avail_profile = GRAPH(time) | |
| 48. | (1.00, 0.21), (2.00, 0.23), (3.00, 0.26), (4.00, 0.29), (5.00, 0.305), (6.00, 0.345), (7.00, 0.39), (8.00, 0.44), (9.00, 0.47), (10.0, 0.525), (11.0, 0.575), (12.0, 0.665) | |
| 49. | network_avail_profile = GRAPH(time) | |
| 50. | (1.00, 0.997), (2.00, 0.991), (3.00, 0.992), (4.00, 0.992), (5.00, 0.993), (6.00, 0.974), (7.00, 0.997), (8.00, 0.991), (9.00, 0.992), (10.0, 0.992), (11.0, 0.993), (12.0, 0.974) | |
| 51. | supplier_avail_profile = GRAPH(time) | |
| 52. | (1.00, 0.28), (2.00, 0.69), (3.00, 0.475), (4.00, 0.38), (5.00, 0.695), (6.00, 0.51), (7.00, 0.66), (8.00, 0.645), (9.00, 0.74), (10.0, 0.6), (11.0, 0.675), (12.0, 0.61) | |
| 53. | testsys_avail_profile = GRAPH(time) | |
| 54. | (1.00, 0.985), (2.00, 0.985), (3.00, 0.83), (4.00, 0.992), (5.00, 0.88), (6.00, 0.99), (7.00, 0.9), (8.00, 0.895), (9.00, 0.992), (10.0, 0.905), (11.0, 0.905), (12.0, 0.905) | |
| 55. | transport_avail_profile = GRAPH(time) | |
| 56. | (1.00, 0.997), (2.00, 0.991), (3.00, 0.992), (4.00, 0.992), (5.00, 0.993), (6.00, 0.974), (7.00, 0.997), (8.00, 0.991), (9.00, 0.992), | |

| | Equation | Description (first occurrence) |
|---|---|---|
| | (10.0, 0.992), (11.0, 0.993), (12.0, 0.974) | |
| 57. | | |
| 58. | ***Sector: Infrastructure-Remote*** | |
| 59. | builtindiag#wt_2 = 0.156 | |
| 60. | builtin_diagnostic_2 = builtindiag_avail_profile_2*builtindiag#wt_2 | |
| 61. | docsys#wt_2 = 0.2 | |
| 62. | doc_systems_2 = docsys_avail_profile_2*docsys#wt_2 | |
| 63. | faultdiag#wt_2 = 0.022 | |
| 64. | faultdiagnostic_systems_2 = faultdiag_avail_profile_2*faultdiag#wt_2 | |
| 65. | helpdesk#wt_2 = 0.178 | |
| 66. | helpdesk_2 = helpdesk_avail_profile_2*helpdesk#wt_2 | |
| 67. | improvement_2 = 1.1 | |
| 68. | infrastructure2_ = improvement_2*((builtin_diagnostic_2+doc_systems_2+faultdiagnostic_systems_2+helpdesk_2+knowledge_base_2+spares_2+supplier_support_2+prev_maint_system)) | |
| 69. | kb#wt_2 = 0.067 | |
| 70. | knowledge_base_2 = kb_avail_profile_2*kb#wt_2 | |
| 71. | prev_maint_system = 0.1 | |
| 72. | spares#wt_2 = 0.111 | |
| 73. | spares_2 = spares_avail_profile_2*spares#wt_2 | |
| 74. | supplier#wt_2 = 0.133 | |
| 75. | supplier_support_2 = supplier_avail_profile_2*supplier#wt_2 | |
| 76. | builtindiag_avail_profile_2 = GRAPH(time) | |
| 77. | (1.00, 0.3), (2.00, 0.31), (3.00, 0.31), (4.00, 0.31), (5.00, 0.505), (6.00, 0.51), (7.00, 0.51), (8.00, 0.51), (9.00, 0.695), (10.0, 0.7), (11.0, 0.695), (12.0, 0.695) | |
| 78. | docsys_avail_profile_2 = GRAPH(time) | |
| 79. | (1.00, 0.2), (2.00, 0.235), (3.00, 0.28), (4.00, 0.295), (5.00, 0.34), (6.00, 0.36), (7.00, 0.385), (8.00, 0.43), (9.00, 0.465), (10.0, 0.48), (11.0, 0.485), (12.0, 0.54) | |
| 80. | faultdiag_avail_profile_2 = GRAPH(time) | |
| 81. | (1.00, 0.2), (2.00, 0.21), (3.00, 0.215), (4.00, 0.205), (5.00, 0.205), (6.00, 0.21), (7.00, 0.21), (8.00, 0.34), (9.00, 0.265), (10.0, 0.345), (11.0, 0.325), (12.0, 0.37) | |
| 82. | helpdesk_avail_profile_2 = GRAPH((time)) | |
| 83. | (1.00, 0.495), (2.00, 0.155), (3.00, 0.9), (4.00, 0.065), (5.00, 0.95), (6.00, 0.2), (7.00, 0.985), (8.00, 0.25), (9.00, 0.93), (10.0, 0.325), (11.0, 0.94), (12.0, 0.69) | |
| 84. | kb_avail_profile_2 = GRAPH(time) | |
| 85. | (1.00, 0.21), (2.00, 0.23), (3.00, 0.26), (4.00, 0.29), (5.00, 0.305), (6.00, 0.345), (7.00, 0.39), (8.00, 0.44), (9.00, 0.47), (10.0, 0.525), (11.0, 0.575), (12.0, 0.665) | |

| | Equation | Description (first occurrence) |
|---|---|---|
| 86. | spares_avail_profile_2 = GRAPH(time) | |
| 87. | (1.00, 0.495), (2.00, 0.705), (3.00, 0.235), (4.00, 0.79), (5.00, 0.295), (6.00, 0.8), (7.00, 0.385), (8.00, 0.795), (9.00, 0.21), (10.0, 0.77), (11.0, 0.305), (12.0, 0.77) | |
| 88. | supplier_avail_profile_2 = GRAPH(time) | |
| 89. | (1.00, 0.28), (2.00, 0.69), (3.00, 0.475), (4.00, 0.38), (5.00, 0.695), (6.00, 0.51), (7.00, 0.66), (8.00, 0.645), (9.00, 0.74), (10.0, 0.6), (11.0, 0.675), (12.0, 0.61) | |
| 90. | | |
| 91. | ***Sector: Onsite Support Personnel*** | |
| 92. | support_specialists_2(t) = support_specialists_2(t - dt) + (up_to_speed_2 - people_leaving_2) * dt | |
| 93. | INIT support_specialists_2 = 5 | |
| 94. | up_to_speed_2 = CONVEYOR OUTFLOW | |
| 95. | o TRANSIT TIME = programme_duration_2 | |
| 96. | people_leaving_2 = turnover_2 | |
| 97. | training_programme_2(t) = training_programme_2(t - dt) + (entry_rate_2 - up_to_speed_2) * dt | |
| 98. | INIT training_programme_2 = 3 | |
| 99. | o TRANSIT TIME = varies | |
| 100. | o INFLOW LIMIT = INF | |
| 101. | o CAPACITY = INF | |
| 102. | entry_rate_2 = people_joining_2*recruitment_policy_2 | |
| 103. | up_to_speed_2 = CONVEYOR OUTFLOW | |
| 104. | o TRANSIT TIME = programme_duration_2 | |
| 105. | productivity_2 = 1 | |
| 106. | programme_duration_2 = 2 | |
| 107. | recruitment_2 = 1 | |
| 108. | recruitment_policy_2 = people_leaving_2*recruitment_2 | |
| 109. | remote_personnel = (support_specialists_2/7)*motivation_2*productivity_2*burnout_impact_2 | |
| 110. | burnout_impact_2 = GRAPH(problem_backlog) | Graphical function. (x-y) |
| 111. | (0.00, 1.00), (90.9, 1.00), (182, 1.00), (273, 1.00), (364, 0.94), (455, 0.88), (545, 0.82), (636, 0.74), (727, 0.66), (818, 0.6), (909, 0.545), (1000, 0.485) | Graph data. |
| 112. | motivation_2 = GRAPH(people_leaving_2) | |
| 113. | (0.00, 1.00), (1.00, 1.00), (2.00, 0.955), (3.00, 0.865), (4.00, 0.8), (5.00, 0.65), (6.00, 0.535), (7.00, 0.49), (8.00, 0.435), (9.00, 0.39), (10.0, 0.38) | |
| 114. | people_joining_2 = GRAPH((time)) | |
| 115. | (1.00, 0.00), (2.00, 0.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.00), (6.00, 0.00), (7.00, 1.00), (8.00, 0.00), (9.00, 0.00), (10.0, 0.00), (11.0, 0.00), (12.0, 0.00) | |
| 116. | turnover_2 = GRAPH((time)) | |

| | Equation | Description (first occurrence) |
|---|---|---|
| 117. | (1.00, 0.00), (2.00, 1.00), (3.00, 0.00), (4.00, 0.00), (5.00, 0.00), (6.00, 0.00), (7.00, 0.00), (8.00, 1.05), (9.00, 0.00), (10.0, 0.00), (11.0, 0.00), (12.0, 0.00) | |
| 118. | | |
| 119. | ***Sector: Problem Sources*** | |
| 120. | problemsources = ((quality&rework)+(gen&process&changes_)+(machine_ageing)+(newprojects)) | |
| 121. | quality&rework = rework%*solution_rate/Problem_level | |
| 122. | rework% = 0.1 | Slider setting. |
| 123. | gen&process&changes_ = GRAPH((time)) | |
| 124. | (1.00, 1.00), (2.00, 0.6), (3.00, 0.9), (4.00, 1.50), (5.00, 1.20), (6.00, 1.20), (7.00, 0.99), (8.00, 1.37), (9.00, 1.01), (10.0, 0.7), (11.0, 1.20), (12.0, 0.85) | |
| 125. | machine_ageing = GRAPH((time)) | |
| 126. | (1.00, 1.12), (2.00, 1.21), (3.00, 1.29), (4.00, 1.05), (5.00, 1.20), (6.00, 1.43), (7.00, 1.07), (8.00, 1.25), (9.00, 1.35), (10.0, 1.23), (11.0, 1.58), (12.0, 1.41) | |
| 127. | newprojects = GRAPH((TIME)) | |
| 128. | (1.00, 0.8), (2.00, 0.9), (3.00, 0.9), (4.00, 1.80), (5.00, 1.62), (6.00, 1.20), (7.00, 0.94), (8.00, 0.93), (9.00, 1.83), (10.0, 1.80), (11.0, 1.02), (12.0, 0.84) | |
| 129. | | |
| 130. | ***Sector: Productivity, Motivation, Burnout (PMB)*** | |
| 131. | motivthresh = 100 | Slider setting. |
| 132. | motiv_drop = 0.8 | Slider setting. |
| 133. | PMB = productivity*burnout_impact*motivation_ | |
| 134. | productivity = 1 | Slider setting. |
| 135. | burnout_impact = GRAPH(problem_backlog) | |
| 136. | (0.00, 0.955), (90.9, 0.94), (182, 0.92), (273, 0.895), (364, 0.9), (455, 0.9), (545, 0.885), (636, 0.61), (727, 0.56), (818, 0.53), (909, 0.49), (1000, 0.475) | |
| 137. | motivation | |
| 138. | motivation_ = IF(problem_backlog>motivthresh) THEN(motiv_drop) ELSE(1) | |
| 139. | | |
| 140. | ***Not in a sector*** | |
| 141. | problem_backlog(t) = problem_backlog(t - dt) + (inflow_rate - solution_rate) * dt | |
| 142. | INIT problem_backlog = 100 | |
| 143. | inflow_rate = incoming +upsets_due_to_improv | |
| 144. | solution_rate = outgoing+product_&_maint_improvements | |
| 145. | central_people_asset = (support_specialists/6)*PMB | |
| 146. | incoming = problemsources*Problem_level | |
| 147. | outgoing = | |

| | Equation | Description (first occurrence) |
|---|---|---|
| | (central_people_asset+remote_people_asset_+infrastructure+infrastructure2_)*Problem_level | |
| 148. | Problem_level = 100 | |
| 149. | remote_people_asset_ = remote_personnel | |
| 150. | solution_contrib = central_people_asset+infrastructure+infrastructure2_+remote_people_asset_ | |
| 151. | upsets_due_to_improv = product_&_maint_improvements*0.4 | |
| 152. | product_&_maint_improvements = GRAPH((TIME)) | |
| 153. | (0.00, 0.00), (1.09, 0.1), (2.18, 0.1), (3.27, 0.3), (4.36, 0.5), (5.45, 0.9), (6.55, 2.50), (7.64, 5.40), (8.73, 10.2), (9.82, 13.0), (10.9, 15.4), (12.0, 16.7) | |