Case Study: Human Resource Management at Simultech

Background

Simultech Pty Ltd[1] is a company that delivers professional consulting services in the information technology (IT) field to a variety of government departments and industry. Recent growth in the government sector has led Simultech's executives to give priority to government work in preference to work in industry. Industry work has generally been focused on shorter-term tasks. Government work tends to be longer term. Simultech is now looking to expand its government work, and the prospect of an expansion of government outsourcing suggests that this is a most viable way ahead.

A discussion has been recorded below involving the following Simultech executives:

- Simon Williamson (SW): Chief Executive Officer
- Todd Davis (TD): Technical Director
- Alistair Jones (AJ): Business Development Manager
- Ron Smith (RS): Personnel Manager
- Janet Brown (JB): Finance Manager

The Executive's Discussion

SW: The recent government contracts have been lucrative for Simultech. I would like to see us continuing to expand in this important sector, particularly in the longer term.

AJ: I agree, but we still have a number of promising industry contracts in the offing.

TD: We must be careful not to depend too much on the individual consultants to work independently. Whichever way we go, I think we need to look at maintaining and developing skills through our mentoring program. At the moment, our planning for mentoring is on the following basis:

- Each expert consultant oversees the technical skills development of four to six experienced consultants, mainly through on-the-job supervision.
- Each experienced consultant oversees the technical skills development of three to five rookie consultants, mainly through on-the-job supervision.

TD: I have been unhappy with the technical quality of some of our recent work. This has led to rework and extra effort, which has cost us money. This leads me to think that in both cases these mentoring ratios should be reduced to around three.

JB: That will drive up our costs and reduce our profits, but we need to look at this.

[1] Further details of this problem and its suggested solution can be found at: http://www.systemdynamicsapplications.com

Prepared by: Dr Alan McLucas
Correct at: 5 July 2005

TD: Well, we have to balance that against the need to do quality work. It is through building a solid reputation for doing quality work that we win repeat work with our customers. Also, I appreciate that reducing the mentoring ratios will place higher demands on our senior (experienced and expert) staff members.

RS: If we win the next phase of the Defence IT Security Contract planned to start in July 2007, I see that we will need to increase our total numbers of senior staff members (experienced and expert consultants) to at least 25. As at today's date, 1 July 2005, the numbers of consultants we have are: 65 rookies, 15 experienced, and three experts.

JB: To increase to those levels by July 2007 will drive up our wages bill. We must be careful not to commit to any new contracts, if we cannot deliver. What you are suggesting could create cash flow problems for us.

SW: Wait a minute … we need some sort of plan for the future …let's not argue semantics!

TD: I agree that we have to be careful here. Remember in 2002 that just after we won the contract with Department of Health, two of our senior consultants left to work for SyInfo, our main competitor.

RS: But, we have managed to keep our personnel losses to low levels. If I recall correctly, over the past few years our personnel losses through consultants leaving have been (in round terms – taking the 2002 situation as an aberration on the figures) – rookies have left at a rate of 12% per annum and experienced consultants have left at a rate of 6% per annum. As you all know well, our turnover is low compared to the rest of the industry. Notwithstanding that, I would be much happier if our losses were even less, but I do not know how feasible that is.

JB: I know it would cost us more money, but could we offer some kind of annual bonus to encourage our best people to stay … that might even save us money in the long term. What about a $15,000 a year bonus to an Employee of the Year, selected by a secret vote by all our consultants?

TD: That might attract me to stay … (just joking) …but we need to think about how a bonus system might work.

SW: Wait a minute! We are skirting around the main issue here. We need to recruit and train sufficient rookies to give us the numbers of senior consultants we need for future work? We are currently committed to the following times in rank before promoting our staff members:

- Recruits: 6 months (training)
- Rookies: 3 years
- Experienced: 5 years
- Expert: not applicable

Further, JB has briefed me that we are about to run into problems with expert consultants. Our expert consultants are getting close to their retirement ages and we expect to lose them at a rate of 50% per year over the next two or three years. I would particularly like to do something to retain these valuable expert staff for some time, to help us over our immediate problem.

TD: The IT schools of the local universities and specialist training delivery companies like Wyzard Informatics Training (WIT) are also ramping up their course throughput in anticipation of the sort of demands we are creating for rookies. Indications at this stage are that we should be able to recruit 15-18 rookies each year for the next couple of years, or so.

RS: Yes, but don't forget that before we can put them to work we have to train them. We can't put any rookie on a job (even with close supervision) until they have completed our own six-month's specialist training course. Given that WIT conduct this training especially for us via a contract we have in place until 2009, at least we are not looking to have additional demands placed on our senior staff to conduct training in addition to their mentoring roles. Our seniors are already over worked: we should be looking to make it easier for them, not harder.

RS: To accommodate this growth in Simultech's contracts, perhaps we might reduce our specialist training course duration or the time-in-rank before we consider promoting … alternatively we might reduce the mentoring demands on our seniors?

TD: Be careful here, I do not want the quality of our technical work to suffer through any of these initiatives. I appreciate the pressures we might be under, but surely our goal is to delivery quality work in the long term.

SW: I agree that quality and repeat work are critical to our future. However, at this stage we need to develop a strategy to get us from where we are now to where we want to be in July 2007. I also want to have assurance that we are not going to over-commit by employing too many staff. I want you all to think about how we do this, and come up with some firm recommendations.

Challenges

The challenges here are to:

a. sufficiently define the problem Simultech faces, and

b. build a system dynamics model to investigate remedial strategies.

Given the short timeframe, you need to plan this as a group model-building project.

Be prepared to recommend a set of strategic options to the executives at their meeting this same time next week.

Prepared by: Dr Alan McLucas
Correct at: 5 July 2005

A SYSTEMS ENGINEERING APPROACH TO SYSTEMS DYNAMICS MODELLING

A Top-down Approach

Systems engineering begins by addressing the complex system as a whole, which facilitates the initial analysis of model functionality, allocation of modelling requirements as well as the subsequent analysis of the module, sector or model and its interfaces. Once system-level requirements are understood, the problem is then progressively broken down into smaller and smaller chunks until the smallest chunks (modules) have been defined and we have a complete understanding of the structure of the problem from top to bottom. This top-down approach is a very important. It enables viewing the system as a whole initially and then progressively breaking the problem down, recognising that structure is important in system dynamics modelling. The interaction between structural elements can be understood more thoroughly, which assists in identifying and designing the necessary (internal) interfaces and between the problem being analysed and the environment (external interfaces).

It must be recognized, however, that while analysis is conducted top down the model can only be built using a bottom-up approach. That is, one major aim of system engineering can be considered to be to provide a rigorous, reproducible process by which the complex system can be broken into a series of simple components that can then be designed and developed using the traditional engineering bottom-up approach. Importantly, the other major aim of systems engineering is to provide a process by which the components and subsystems can be integrated (synthesised, in systems engineering terms) to achieve the desired system properties (and system dynamics structural behaviour).

Integration aims to combine lower-level modules and sectors into progressively higher-level sectors and models until the overall model is complete. While the design process has been conducted top-down, the integration process is conducted bottom-up using well-proven techniques. At each stage of the integration (synthesis), some form of integration testing is conducted to verify the successful integration against the appropriate level of documentation. Eventually, when systems integration is complete, testing can be conducted at the system level against the original modelling requirements. The integration (synthesis) effort is summarized in Figure 1. Note that the terms system (model), subsystem (sector) and component (module) are relative.

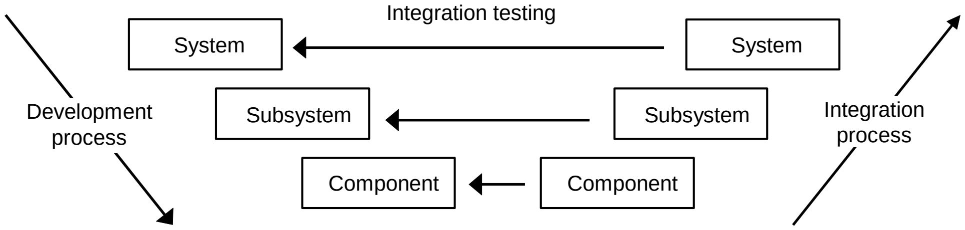

Figure 1. Top-down development and bottom-up integration process.

A More-Rigorous Approach to System Dynamics Modelling

Systems engineering, therefore, has much to offer system dynamics modelling. In particular, systems engineering offers a framework through which design and development discipline is applied and rigour assures reliable outcomes. System dynamics models bring together hard

Prepared by: Dr Alan McLucas
Correct at: 5 July 2005

and soft aspects that are difficult to evaluate and test (verify) in detail, but to assure the necessary rigour and opportunities to learn, comprehensive testing is essential.

Arguably, the challenges faced by system dynamicists in verifying their models (that they work 'right' in accordance with requirements) are greater because of the influence of soft variables and the variability of human behaviour in human activity systems and human response to changes (changes applied through exogenous forces or those which develop within the human activity system). This places greater demands on the modeller to be able to design and apply tests that verify that the models actually reproduce the cause-and-effect relationships of the real-world problem situation.

System dynamics models must explain real-world behaviour through structure and equations that reflect real causal relationships as they appear in the real world system (Forrester, 1961: 115-129). These models must be tested to assure that they are the 'right' models (validated) to provide the detailed insights needed for the design and development of appropriate remedial strategies. So, discipline and rigour in system dynamics modelling is essential.

Any model-building activity must be managed as a project designed to deliver specific outcomes. These outcomes include learning through experimentation with the model and delivery of a completed model. The project must be based on *requirements*:

- *elicited with close engagement of key stakeholders,* notably the client group for whom the model is being built;
- *analysed for logical construction and completeness*—here conflicts between requirements must be resolved whilst maintaining a minimum set of requirements which must be met (built into the model) according to priority;
- defined, that is, *clearly and unambiguously specified using precise language*;
- *validated*, that is, tested to determine the extent to which the requirements are likely to lead to development of a model which maps sufficiently onto the real-world problem space;
- managed throughout—here the *modelling project* must be *managed in terms of* its:
  - *scope*, particularly as the meanings of modelling requirements (interpretations of dynamic hypotheses) are questioned, which often leads to pressure to change the scope of the modelling effort ; and
  - *configuration*, that is, what form the model will take and what it will include or exclude must be constantly monitored through a set of formal processes, especially where the allocation of effort to tasks involves potential overlap and confusion;
- *traceable through each step*, including:
  - *back to the original requirements* (and testable, that is, verifiable against the original requirements); and
  - *changes* being *controlled through configuration management*.

The system dynamics modelling process is illustrated in Figure 2 (adapted from Forrester, 1994: 245), annotated to show where systems thinking and soft systems methodology and systems engineering activities are integrated.

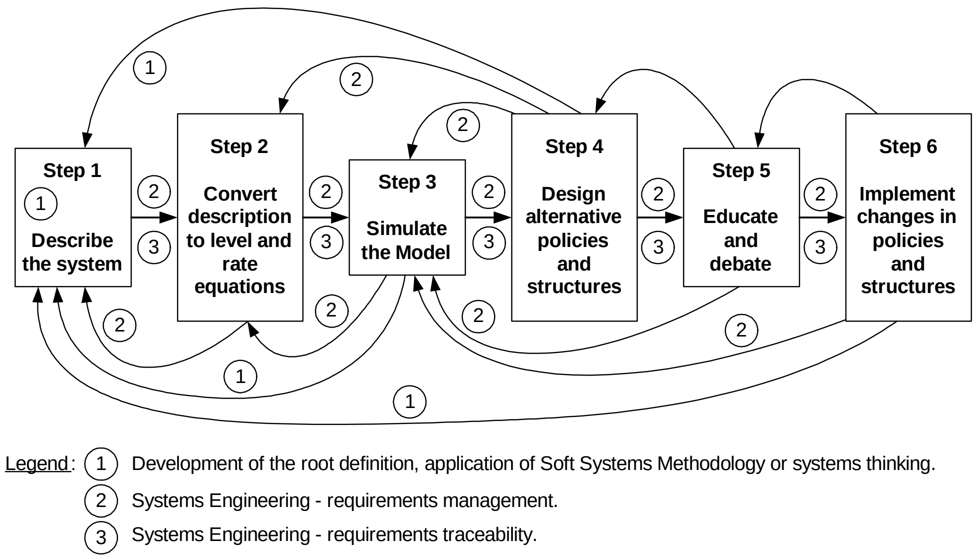

Figure 2. System dynamics modelling process annotated with Soft Systems Methodology, Systems Thinking and Systems Engineering activities.

A Top-down Approach to Coping with Complexity

Systems engineering begins by addressing the complex system as a whole, which facilitates the initial allocation of requirements as well as the subsequent analysis of the system and its interfaces. Once system-level requirements are understood, the system is then broken down into subsystems and the subsystems further broken down into components until a complete understanding is achieved of the system from top to bottom. This top-down approach is a very important element of managing the development of complex systems. The approach leads to clear definition of the component parts (modules), sectors and co-models and the interfaces between them (McLucas and Ryan, 2005). A generic diagrammatic representation of a module is the lowest-level *physical* component part, is shown in Figure 3 (McLucas, 2005).

Group Model Building – Using Functional Break-down to Aid Management

In any model-building project it is important to focus attention first on *functional break-down* and developing *functional definitions*. For a population problem, such as that described by Sterman (2000: 285-289), the functional break-down to modules is shown at Figure 4.

A functional break-down and set of functional descriptions enable the manager of a group model building project to allocate tasks to group members and avoid confusion, particularly with respect to the interfaces between elements of work.

This population example, Figure 4, considers how multiple feedback mechanisms, some including non-linearities (though that is generally not known at this early stage), can produce complex behaviour in populations where the environment has limited capacity to carry a population. From a process point of view, requirements and design are approached top-down (with Figure 4 being an early artefact of the process) but detailed construction follows a bottom-up approach (once each physical module has been described, such as shown in Figure 3). Note that in the process, Figure 4 (*functional architecture* of the model) would be developed before Figure 3 (*physical architecture* of the model).

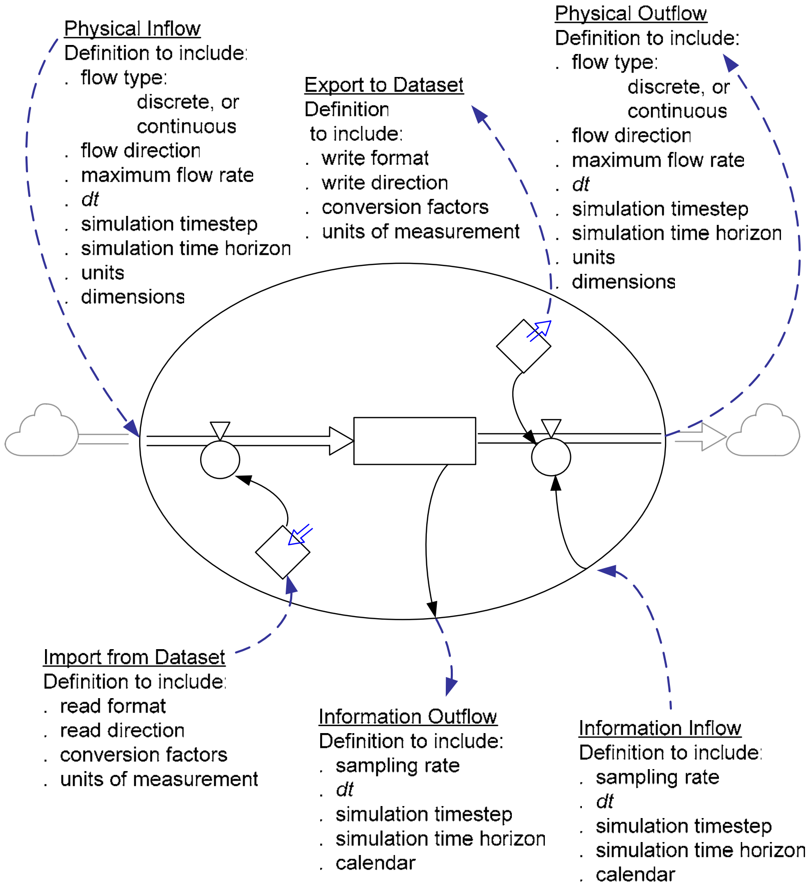

Figure 3. Generic System Dynamics Modelling Module – A *Physical Definition*.

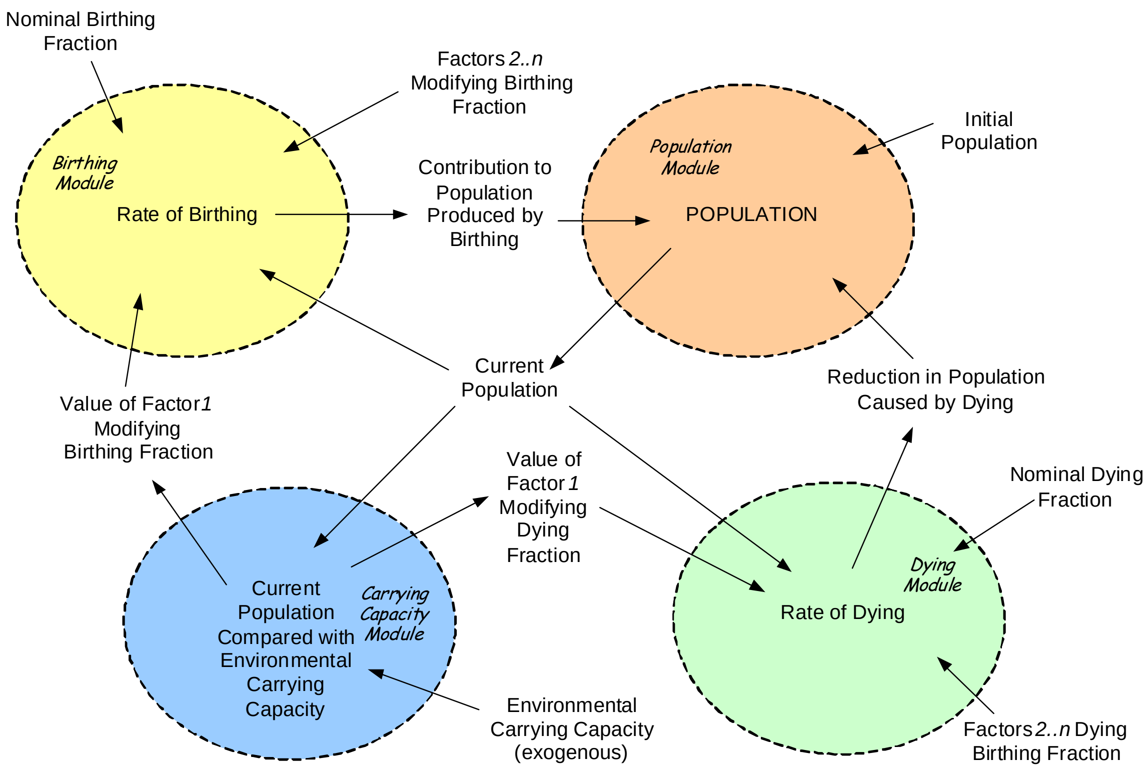

Figure 4. Population Problem – Functional Break-down in Four Modules

The subsequent building of each module, verification and their combining (through integration and synthesis) must be managed through systems engineering methodology, which has the rigour and discipline to assure that none of the system's functionality is lost and emergent properties are systematically discovered. The systems engineering approach also assures that processes of analysis, design and construction can be reproduced and can be implemented in a way that still enables use of the traditional bottom-up approach. From a group model-building viewpoint, it also enables the allocation of modelling effort to building, testing and subsequent synthesis in a way that avoids duplication of effort, misnaming of variables occurring at the interfaces between modules. Consequently, once functional requirements and modules have been defined, group model building becomes a routine matter.

Top-down analysis creates the framework within which bottom-up construction of modules and sectors and ultimately integration into models can then occur. This approach is facilitated by formulation of a clearly defined systemic structure which will lead to creation of models exhibiting the necessary system-level behaviour, that is, models which replicate the reference modes of behaviour (in systems engineering terms, delivers required functionality).

The systems engineering approach also facilitates management of construction of the model in its most-highly aggregated form by enabling the group modelling leader having the overall responsibility for model construction to reflect upon the known behaviour of building blocks of structure.

Feedback loops of archetypical system dynamics structures will frequently exist across the boundaries of the functional modules defined for the group modelling activity. Whilst the group modelling leader will be able to seek out instances of archetypical structures, he or she will be able exploit these structures by coordinating their linkage across the architecture and at each of the defined module / sector / co-model boundaries. The systems engineering approach has the added advantage that it routinely enables the discovery of those feedback mechanisms (and the emergent properties with which they are associated) even when those building individual modules, say as described in Figure 4, do not know explicitly that inputs or outputs to the module they are currently working on are part of a feedback loop.

To manage complexity, therefore, we need to be able to specify the requirement for our model, allocate these requirements appropriately to sectors and then, within sectors, allocate requirements to modules. Having developed and tested the modules, we integrate them into sectors, test the sectors against the subset of requirements, integrate sectors into the model and then test against the system-level requirements.

Emergent Properties

Traditional problem solving involves working from the bottom up. The bottom-up approach assembles well-known, well-understood and manageable components into subsystems. Emergent properties therefore cannot be predicted solely by looking at the components (Stevens, et al., 1998: 94). A bottom-up approach does not deliberately enable the discovery of emergent properties. In system dynamics modelling, analysing reference modes of behaviour as part of a top-down approach (consistent with systems engineering) systematically aids discovery of one particular form of emergent property. This is complex dynamic behaviour produced by feedback and delay. Without this aspect of methodology, the emergent properties and the real causes for them being produced may remain undiscovered.

Verification and Validation

In engineering there is a very strong link between the model and causal explanations underpinning the model, as evidenced by physical laws and the emphasis placed by Engineering Faculties on studying those physical laws. In systems engineering it is expected that these causal explanations can be 'proven'. In system dynamics modelling, despite the warning provided by Forrester (1961: 115-129) that we must explain real-world behaviour through the structure and equations which reflect the real causal relationships in the real-world system, examples of system dynamics models which mimic the real world (but for which there is no real 'proof') can be found.

System dynamics models frequently contain multiple (and non-linear) feedbacks which readily elude our human cognitive capability—where multiple feedbacks exist, we need to be able to develop comprehensive tests to assure that our models behave as they should. The problem that this presents to modellers is that, if we do not have the cognitive capability to understand the feedback mechanisms, how can we know that the tests we design and implement actually verify that our models work as they should?

We can improve our system dynamics modelling by use of molecules of system dynamics structure which we study in detail. Knowledge of these molecules and their behaviour can augment strategies for model testing, if we acknowledge that they can be constructed and tested separately then progressively combined to produce a model whose behaviour is compared with the model we have constructed through the top-down approach described. Without systems engineering and the rigour it brings through progressive verification, we cannot expect to build sophisticated models of complex problems and have those models work properly.

Systems engineering as it is applied to system dynamics modelling can be described as a sequential process of requirements creation and integration into a model following the arrows in Figure 5, (adapted after Forsberg, et al., 2000: 116). The development of the system dynamics model follows the sequence from the top left of the 'Vee', to the bottom, then back up to the top right.

For our purposes, *verification* can be defined as (Jones, 1996: 94-96; Rakitin, 1997: 51-66):

> The process of determining whether or not the products of a given phase of the system dynamics modelling development cycle fulfil the requirements established during the previous phase.

Another way to view verification activities is that verification helps us answer the question: "Are we building the model right?'

In system dynamics modelling, verification is all about ensuring that the governing business rules have been correctly coded, that the structure in which those rules operate results in correct replication of the reference modes of behaviour identified in an earlier stage (and specified as requirements for the model we are building).

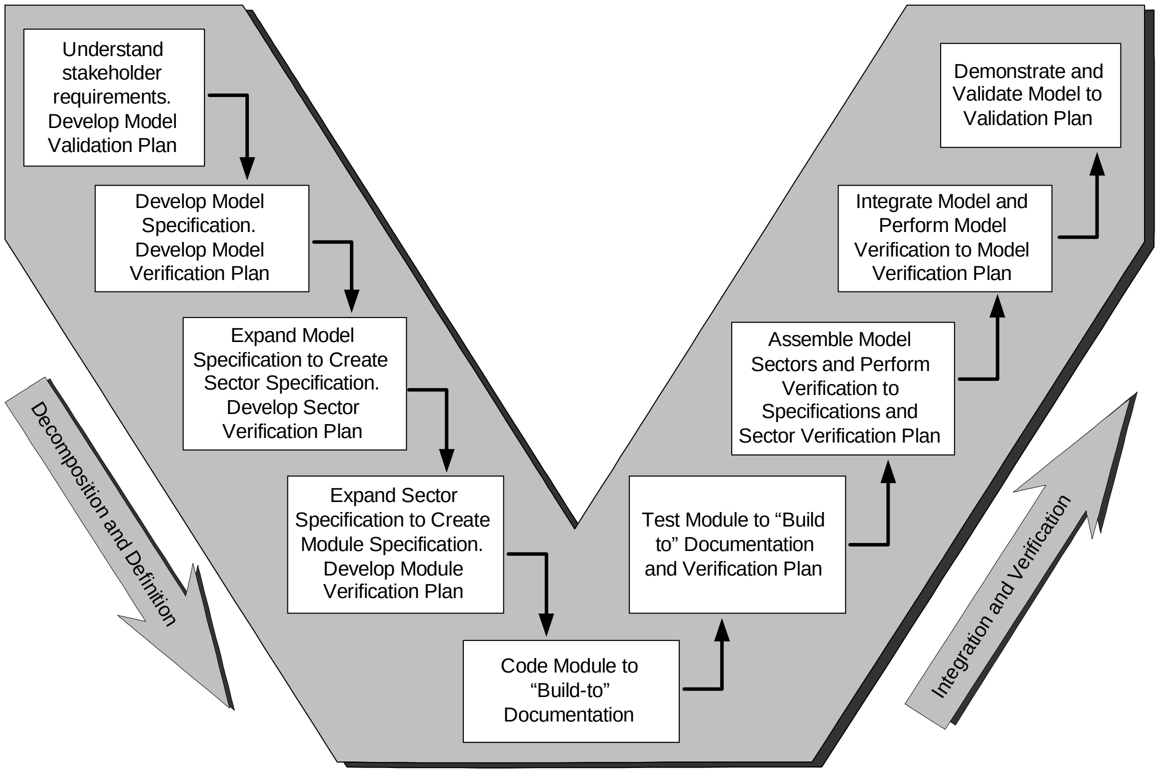

Figure 5. The basic systems engineering 'Vee' model applied to system dynamics model building.

<u>References</u>:

Forrester, J.W., 1961, 'Industrial Dynamics', Productivity Press, Portland, Oregon.

Forrester, J.W., 1994, 'System dynamics, systems thinking and soft OR', in: System Dynamics Review, Vol. 10, No. 2-3, (Summer-Fall): 245-256.

Forsberg, L., Mooz, H. and Cotterman, H., 2000, 'Visualising Project Management', 2nd ed., John Wiley & Sons.

Jones, C., 1996, 'Software defect-removal efficiency', in: IEE Computer, Vol. 29, No. 4, April: 95-96.

McLucas, A.C., 2005, 'System Dynamics Applications: A Modular Approach to Modelling Complex World Behaviour', Argos Press, Canberra, Australia.

McLucas, A.C. and Ryan, M.J., 2005 (forthcoming), 'Meeting Critical Real-World Challenges in Modelling Complexity: What System Dynamics Modelling Might Learn from Systems Engineering' in: *Proceedings of International System Dynamics Conference*, System Dynamics Society, Boston, MA.

Sterman, J.D., 2000, 'Business Dynamics: Systems Thinking and Modelling for a Complex World', Irwin McGraw-Hill.

Stevens, R., Brook, P., Jackson, K. and Arnold, S., 1998, 'Systems Engineering: Coping With Complexity', Prentice Hall, London.

<u>Prepared by</u>: Dr Alan McLucas
<u>Correct at</u>: 5 July 2005

SOLVING SIMULTECH'S PROBLEM - A SYSTEMS ENGINEERING APPROACH TO BUILDING THE SYSTEMS DYNAMICS MODELLING

A Top-down Approach

A top-down, conceptual view of Simultech's problem might be as depicted in Figure 1.

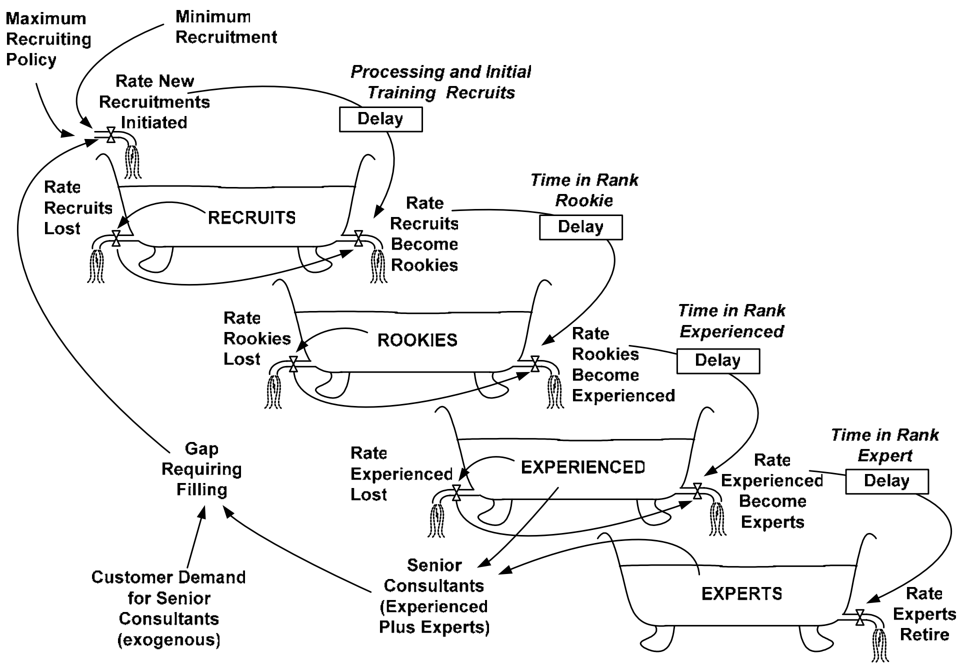

Figure 1. Top-Down Conceptual *Physical* View of Simultech's Problem.

A functional break-down depicting selected modules is shown in Figure 2.

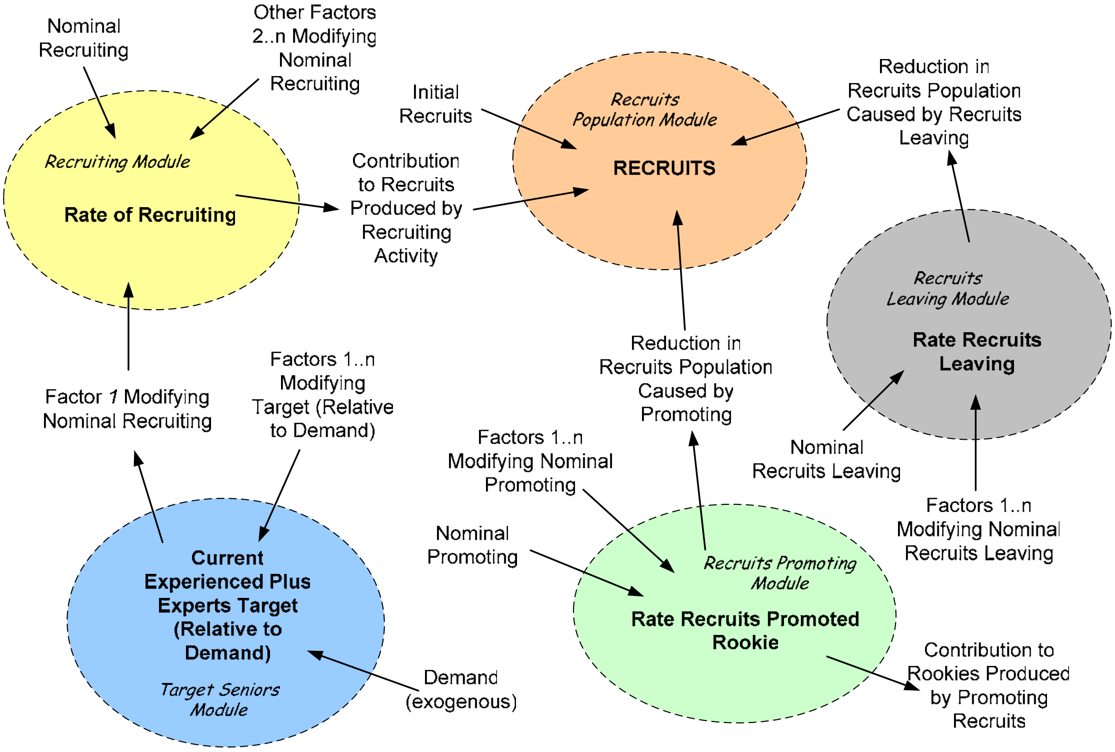

Figure 2. *Functional Break-down* Depicting Selected Modules.

Prepared by: Dr Alan McLucas
Correct at: 5 July 2005

This might be alternatively represented in physical terms as an influence diagram as shown in Figure 3, or as a stock and flow diagram as shown in Figure 4.

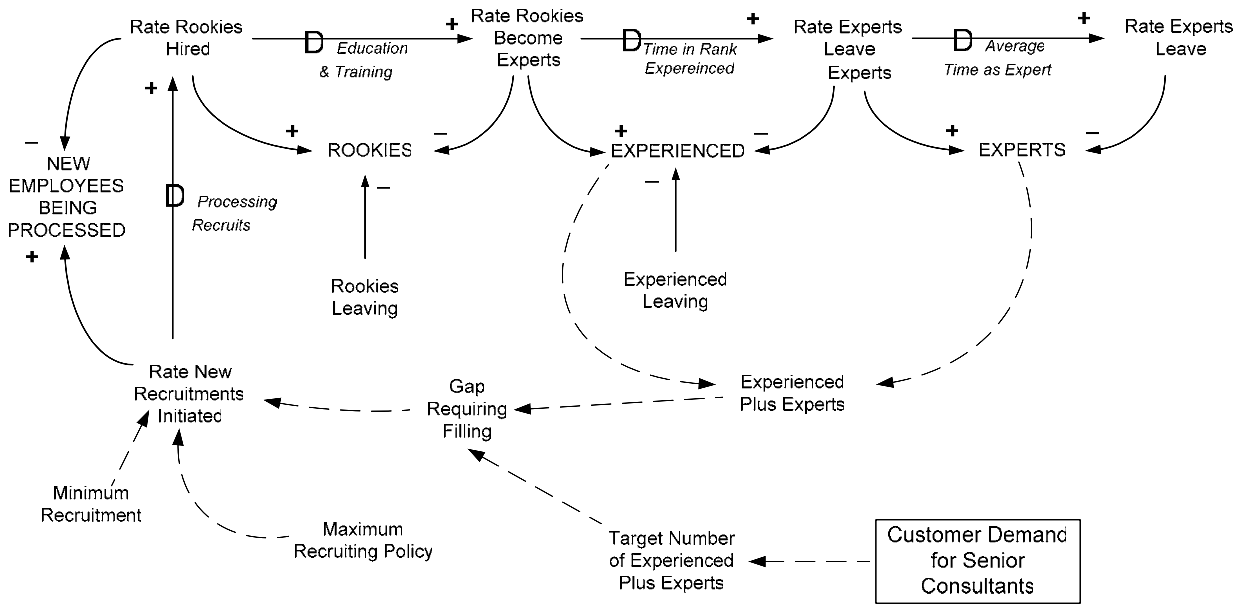

Figure 3. Top-Down View of Simultech's Problem – *Physical* Model Depicted as an Influence Diagram.

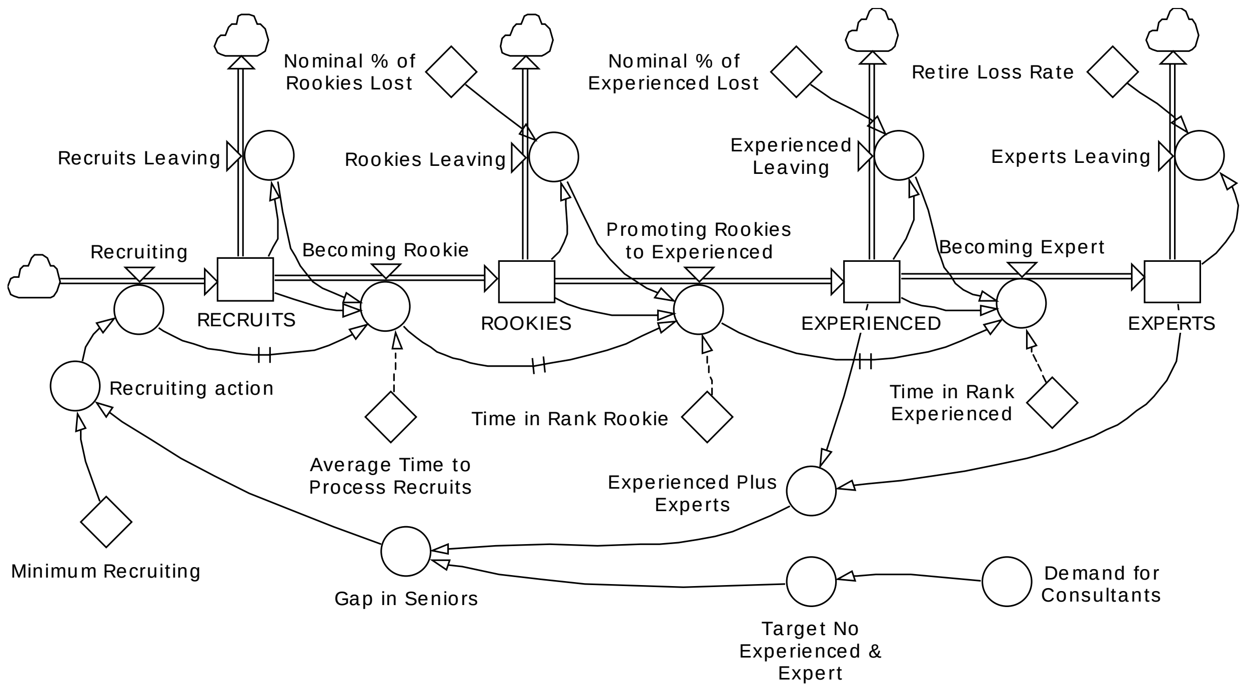

Figure 3. Top-Down View of Simultech's Problem – The *Physical* Model Depicted as a Stock-and Flow Diagram .

<u>Prepared by</u>: Dr Alan McLucas
<u>Correct at</u>: 5 July 2005

System Dynamics Modelling - Modules

The basic physical module upon which the Simultech sectors and model will be built is shown at Figure 4.

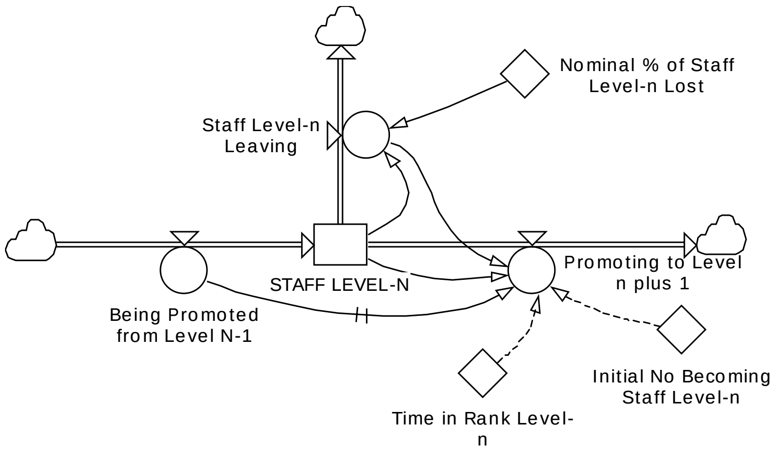

Figure 4. Simultech's Problem – Basic *Physical* Module.

Variable definitions for key are summarised at Table 1.

Variable Name	Definition
Staff Level-n Leaving	'STAFF LEVEL-N'*'Nominal % of Staff Level-n Lost'
Promoting to Level n plus 1	IF('STAFF LEVEL-N'>0<<staff>>,1,0) *DELAYPPL('Being Promoted from Level N-1', 'Time in Rank Level-n', 'Initial No Becoming Staff Level-n') -'Staff Level-n Leaving'
Initial No Become Staff Level-n	Comment: Initial value depends on the need to remove transient effects of 'filling the pipeline'

Table 1. Key Variables of Basic Module.

Prepared by: Dr Alan McLucas
Correct at: 5 July 2005