HPC-Provisional ISBN: 978-0-907833-28-4

Improving communications within systems groups

Hector McNeill
SEEL

Systems Issues Workshop

Decision Analysis Initiative 2010-2015

Portsmouth

August, 2014.

Improving communications within systems groups

Hector McNeill[1]
SEEL

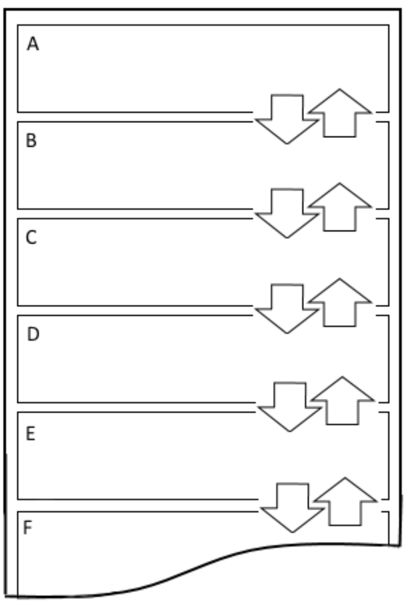
Introduction

This short paper is to introduce a simple concept that formalises what most systems engineers do when approaching complex systems design. That is, we apply the “divide and rule” method of breaking down an apparent complexity into a series of interconnected simpler components. Usually, as in chip design, one component, or object, deploys a method that has an output that represents the input to another object whose methods transforms that into an output to another object and so on.

The simple concept I refer to is a way of visualising a complex system that makes use of an information system in some way and a core data set to sustain the system’s activity. The data sets used are made explicit so as to make the data collection/input method apparent. For want of a better term I refer to this as a Data Reference Model (DRM). The whole process can be analysed in a transparent way by considering the process to be a data model with each process component having different but distinct contributory functions in the handling of the data. Thus the method to be deployed in a Data Reference Model where the word “reference” has the precise usage as being the relation between objects in which one object designates, or acts as a means by which to connect to or link to, another object. The first object in this relation is said to refer to the second object. The connection between these objects, or different parts of a process, is the data, received as input and output in some altered form. Thus the term, Data Reference Model.

DRM Data Reference Model

I used a little bit of OO terminology in the introduction. Very often we are looking for a Decision Analysis Model as a basis for establishing some determinate structure which, preferably, can be used to simulate a complex system design in order to test if we have a sound understanding of the series of interrelated functions that generate outputs that should fall within expected ranges. If we design a system correctly we end up, in most cases, with a cascade of a need-solution logic. Thus what happens in one object needs inputs from another object which in turn needs inputs from some other object. Each object would have outputs, that are the inputs to the next object. So each object and its method can be represented as a level in a ladder. This is illustrated on the left in its simplest form. As long as the layer at the bottom provides the required inputs to the level above then the model is helping divide the “problem” into component parts as required. If any level is occupied by a more complex object and methods this will usually cause lack of clarity and often this is resolved



[1] Hector McNeill is Director of the Systems Engineering Economics Lab

by dividing that object concerned into two or more derived objects and their associated methods. In programming terms one can break the DRM down to the level of primitives, in the sense being the smallest 'unit of processing' available to a programmer. The important issue to grasp is that what is flowing through the system as outputs and inputs is data streams which are altered by each successive layer, level or object. I hope that we would accept that if DRMs are based on this ability to divide object-methods into primitives or to integrate them into distinct algorithms, combining a set of primitives, then it becomes evident that, theoretically, at least, DRMs can handle any level of complexity. I also hope that those of you who use any of the many IT system modelling languages will realise that a DRM can overlay most known IT language models. This is of fundamental importance in systems groups as a basis for communicating what they are planning to do, are doing or have done to management or oversight groups. In recent encounters organized by DAI, the general conclusion was that many IT modelling languages are too complex for non-specialists and cannot be used effectively for communication for most decision makers or clients because of the model specific terminology deployed. One example is UML (Unified Modeling Language) adopted by the Object Management Group in 1997 and even made an ISO standard in 2005[2]. However, this system is seldom used in industry because it is so cumbersome and, in spite of the claims, many non-IT people find difficulty comprehending the flow across multiple documents instead of a neat summary. Many consider the “Waterfall model” as a well-established transparent method for developing software. However, many deploy this as a unidirectional no-feedback system while others include re-iterative feedback loops. There does not appear to be any formal generic method that can be picked up and deployed with ease.

After some 40 years working in systems engineering working with groups involving between 10 to 200 participants, it has become very evident to me that clarity of purpose of design and communication of solutions is best achieved on the basis of logic and plain (free of jargon) unambiguous language (plain English in our case) and the use of some easily understood documented representation.

As systems engineers we need to be always concerned with the imperative of understanding what a decision-maker wants and to use means of communication and reporting back systems options that address these objectives, selection criteria and decision-maker preferences. We should never forget that sometimes the decision-maker does not always know fully what he or she wants, however, we also should guard against assuming that they might not know what they want. This is why we have a job to do in making explicit levels of understanding and intent. It is often the case that good systems engineering will identify objectives and solutions that do not comply with pre-determined preferences but prove to be far superior. Systems engineering is after all a learning process for all concerned.

Thus data elements become the smallest discrete objects in the sense that each data set represents a specific property in the specific context within which it is used. Therefore, the binary code 001001101 will signify what a specific subsystem or object method assumes it to mean. Non-the-less all through the system different combinations of binary codes flow up and down a system and the significance of each code is understood by the methods associated with each of the objects in a data flow or chain. This is an important topic but it does not help my explanation of where I want to guide this presentation. It contains many possible

[2] ISO/IEC 19505-1:2012 Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 1: Infrastructure

diversionary elements, and very interesting discussions concerning such things as semantics which, for the purposes of this presentation, I want to avoid.

Significance

Rather than pursue this somewhat esoteric explanation I want to turn to the two main practical applications of the DRM. These are:

- Designing, describing, analysing and communicating the details of complex systems
- Building decision analysis models to better simulate and design complex systems

In systems engineering we are concerned with finding solutions to a wide range of issues that concern communities, corporate managers, labour unions, engineering designers and government agencies. The objective is to identify the most economic, effective and efficient solution to an identified issue. So the objective becomes the satisfaction of a need usually expressed in terms of a gap in current provisions. So, for example, the gap could be quantified in terms of the desire to attain an improved status for a target population, such as a higher real income, improved health status or higher corporate profit and cash flow. There might be a need recognised by the fact that, somewhere, a known specific technology and applied techniques achieves a desirable level of productivity. The current attainment of local systems might be below this so the gap to be filled is the measured difference.

Now to achieve this in an effective manner, which helps bring some sustainability by means which accommodate future eventualities, we need to avoid analytical solutions that provide no structure or do not use some form of decision analysis model. Whoever is "left" with a solution, needs to have the means to sustain its evolution in a desirable direction. Indeed, the reason Decision Analysis evolved as a distinct discipline[3], under the pioneering work of Ronald Howard, was his desire to create a logical method that made sure all relevant factors and possible eventualities had been taken into account. So the first step in designing, describing, analysing and communicating the details of complex systems is to have a transparent means of initiating this process; the DRM has an important role here.

Moving on to the construction of a decision analysis model to simulate complex systems would be the result of an initial DRM screening. Indeed, I mentioned the need-solution object - method, input-output, orientation above to maintain the awareness that the DRM can end up as the structure of a decision analysis model associated with data flow and which can be programmed to simulate the process outlined in the DRM.

This was the basis for the successful development of the Seel-Telesis program[4] in 1989 as a decision support system for corporate decision-making using the Clarion language, which was not in fact object oriented. This worked because it was based on a structural production function approach or structured input-output model.

Why simulate?

[3] McNeill, W.W., "*The State-of-the-Art & Future of Decision Analysis",* GBI, London, HPC, 2009, ISBN: 978-0-907833-21-5.

[4] This was the basis for the successful development of the Seel-Telesis program in 1989 as a decision support system for corporate decision-making using the Clarion language, which was not object oriented. This worked because it was based on a structural production function approach or structured input-output model. This program was designed and implemented with the financial support of the Manpower Services Commission of the Department of Trade & Industry of the Government of the United Kingdom.

Simulation is often perceived to be a theoretical construct, which it is. However, a simulation model also represents out best understanding, and emulation, of an existing or proposed operational system. If our model does not reproduce that is observed in reality by generating improbable outcomes, clearly the model needs to be improved because it reflects our current level of understanding of the problem to be solved. If we are unable to design new solutions based upon a validated model, then we have no way of providing any credible guidance to customers, based on the model. The model should have a durable purpose of ongoing and future analyses to support decision-making in such things as change analysis related to the future sustainability and increasing performance of an implemented system.

ISO Process Approach

Combining the DRMs with the conventional Decision Analysis Model's reiterative cycles (Decision Analysis Cycle) it is possible to identify, for example, an inappropriate layout of a factory floor or allocation of personnel to administrative duties in some government agency, supported by an information system. This can be more easily analysed using a simulation model and the appropriate changes identified, analysed and ranked according to criteria.

The ISO Process Approach[5] (ISO 9000) is a basic approach to this topic but it lacks methodologies such as Decision Analysis and DRMs. The main point about the ISO "model" is the reiterative steps of:

- assess
- deploy
- monitor
- re-assess
- start again!

What is missing is the simulation used to minimise risk by carefully analysing likely decision outcomes before resources are committed, sometimes with potentially severe economic consequences[6]. If I might elaborate, it is both potentially wasteful and risky to commit resources on any system without first of all simulating its functionality from the standpoint of desirable quality of output, general efficiency and costs, ease of operation in terms of a customer's existing resources (plant and human resources), timeliness, and risks. Here we should not forget Howard's simple definition of a decision as an irrevocable commitment of resources to a set of actions. If a decision is changed, this invariably requires the commitment of additional resources or loss of effect of resources already committed. Decisions are important! This is why investing in good decision analysis is of importance.

The future sustainability and evolution in performance of the systems we deliver to our customers is an essential aspect of systems engineering. For example, whereas training of individuals may be part of the systems commissioning exercise to ensure that people can manage an information system, the actual substantive impact of training on an organization depends upon the gaining of more practical operational experience. If such individuals can continue to work on the system they will descend the learning curve to become more skilled at operations based on their individual accumulation of tacit knowledge. As a result, the quantifiable impacts of training become measurable sometime after the completion and delivery of a systems engineering assignment. In order to secure a substantive impact, it is

[5] ISO/TC 176/SC 2/N 544R3, "*The Process Approach*", ISO 2008.
[6] I understand that the ISO is adding the Process Approach to their Quality Management series 9001 but as far as I am aware here is no substantive design/simulation phase included.

necessary for the managers of the system developed, to understand the distinction between tacit and explicit knowledge and to restructure their operations by taking these into account. The simulation model, or failing this, the assignment DRMs, can be used to support the evaluations and decisions taken on such human resources strategies.

We all know that experienced shop floor engineers, older farmers or policy makers make less mistakes because of their accumulated tacit knowledge of what will work “in their environment”. However, significant mistakes can be made when an attempt to introduce new technologies or technique to gain performance through some innovative advances. Here, tacit knowledge needs to be supported by new explicit knowledge, that is, facts and figures on the performance of “other” resource configurations used in similar circumstances in other companies or government administrations. Here the simulation model is of significance in assisting more experienced decision-makers pose their practical questions based on their knowledge of their operational systems and their assessment of the credibility of the options generated by the model.

I cannot over-emphasize the need for people with practical experience and who have:

- descended a learning curve, associated with:
  - a very specific technology
  - applied techniques
- perceptions of the personnel working on the linked operations based on this specific experience

Such people need to have access to precise explicit knowledge on how any new configurations of existing technologies, or new state-of-the-art technologies, have been demonstrated to operate in terms of costs, timeliness, yield (lack of errors) and general effectiveness and efficiency.

Here again the DRM has a role to place in making explicit understanding and helping support a better appreciation of what any decision analysis model is doing.

Last but not least

I started off this presentation discussing the OOP characteristics of the DRM. A clearly specified data set provides as the basis for securing the necessary information to manage a proposed system. The DRM data transformations and algorithms identified, as well as the environment of data collection, input and data access by managers and other users for ongoing monitoring and decision-making, provides a transparent structural context. Such a description of required data manipulation requirements provides useful information to support specification work of the more technical information management system.