BURGE HUGHES WALSH

The Systems Thinking Tool Box

Dr Stuart Burge

".. bump, bump, bump, on the back of his head. It is, as far as he knows the only way of coming downstairs, but sometimes he feels that there really is another way, if only he could stop bumping for a moment and think of it."

Winnie the Pooh - A. A. Milne

Sequence Diagram (SD) Alias Process Model, Functional Model

What is it and what does it do?

A Sequence Diagram[1] is a schematic model of the system of interest that defines the sequence of activities, functions or process steps that convert the inputs into the outputs. It allows a team or an individual to produce a high-level diagram of an existing or planned system that not only defines the boundary of the system of interest but the necessary internal system functionality. Figure 1 shows a Sequence Diagram for a domestic Automatic Washing Machine.

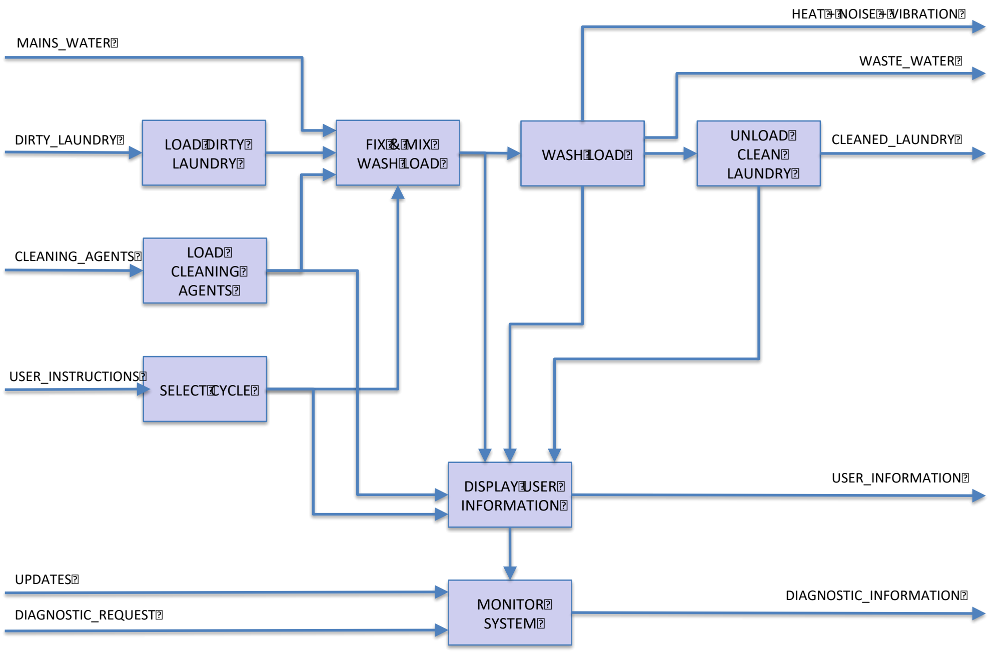

Figure 1: A Sequence Diagram for a Domestic Automatic Washing Machine

[1] The Software Engineering community uses a modelling language called UML (Unified Modelling Language) which has a modelling tool called a Sequence Diagram. While similar to the what is described here it is different and could be a source of confusion

In Figure 1 time flows from left to right and therefore the inputs are on the left hand side and the outputs on the right. The grey boxes are the functions/activities of the system. The connecting arrows show how the basic functionality has to logically cooperate in order the transform the system inputs into the system outputs. In this case dirty laundry into cleaned laundry.

Why do it?

Sequence Diagrams have many uses:

- For new systems they can help:
    - Understand the operational concept through identifying the necessary system functionality.
    - Assess potential emergent behaviour by consideration of how the identified systems functions could fail.
- For existing systems they can help:
    - Capture and document the current “as is” state of a system. This is particularly useful for process intensive system and repetitive Human Activity Systems.
    - Identify system weakness. In particular it is possible to look for activities/functions that are non-value adding (waste) and therefore could be removed increasing efficiency without impacting effectiveness.

When to use it?

A Sequence Diagram is particularly useful in:

- understanding and engineering requirements for a new system.
- analysing an existing system.

They are, in general, relatively simple to construct and can help scope the project by defining the boundary of the system through consideration of the high-level functionality. In essence, the Sequence Diagram defines the scope the system to be designed or analysed. It therefore provides a simple understandable pictorial representation that can be used to obtain and document agreement about the scope of a project or problem.

Who does it?

An individual or a team can construct a Sequence Diagram. Whether it is team or individually based depends on the problem being tackled and the phase of system development.

How to do it?

Before explaining how to construct a Sequence Diagram, it is important to understand the diagramming conventions that are used. There are many varieties of Sequence Diagram, but all show the activities, steps, and functions of the system in the order in which they perform to achieve the purpose of the system. The basic conventions are shown in Figure 2 where the system functions (activities, process steps) are show in the rectangles. System inputs are labelled arrows going into the function boxes. System outputs are the labelled arrows coming out of the function boxes. The unlabelled arrows show the dependencies between the functions. Sequence Diagram has a time order with time increasing from left to right.

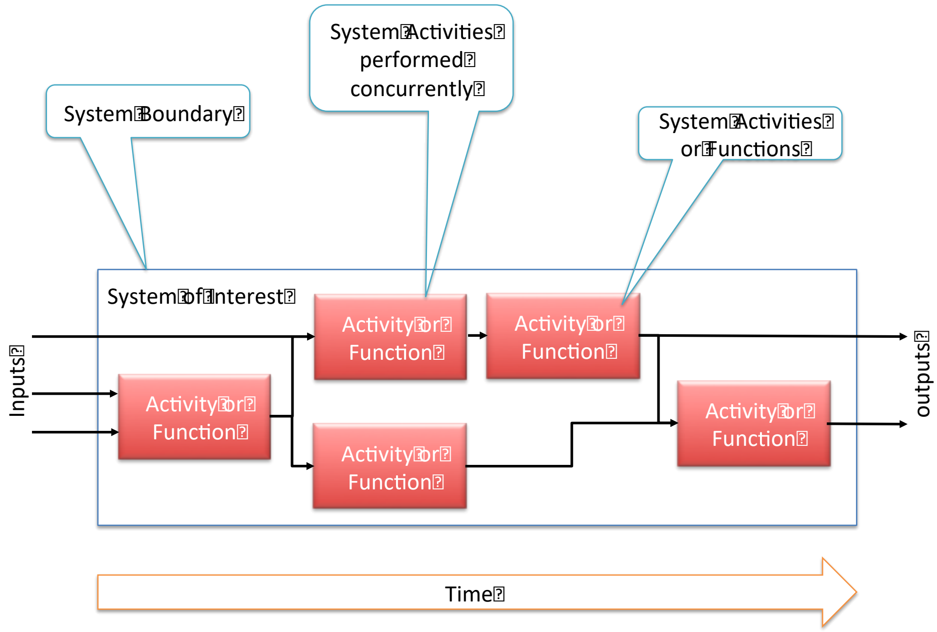

Figure 2: the Basic Sequence Diagram Diagramming Conventions

A key aspect of constructing a "good" Sequence Diagram is to ensure each Activity or Function is defined as a verb noun combination (an action on an object) e.g. Receive Order, Check Stock, Analyse Data etc. It is critical when naming function to ensure that the verb is imperative and the noun is concrete. It is important to avoid abstract nouns (nouns that cannot be touched, felt, smelt or tasted like reliability, safety, style). Imperative verbs are command type verbs. Appendix A gives a list of suitable verbs and guidance on defining functions.

There are lots of additions that can be made to the basic diagram including:

- Naming the "flows" between activities or functions.
- Separating out different types of inputs (control inputs, information inputs, resource inputs etc).
- Including timing details .
- Different symbols for different types of activity.

- Include swim-lanes (very useful for process- or service-based systems).

My experience when constructing Sequence Diagrams to explore a system of interest is to start simple and further detail later.

As noted earlier there are two distinct situations when constructing a Sequence Diagram that alter the approach. They are:

- Existing System
- New System

The basic process for constructing Sequence Diagrams in these situations are as follows;

New System:

1. Assemble a team of experts.
2. Define clearly the purpose of the system.
3. Follow logically the main inputs to the system and determine what has to happen to (identify the functions) them to deliver the output.
4. Examine the model for potential issues (failures, interactions, omissions etc) to uncover any undesirable emergent behaviour.
5. Determine additional functions to mitigate undesirable emergent behaviour.

Existing System:

1. Assemble a team of experts.
2. Define the scope of the system under investigation.
3. Define the purpose of the system under investigation.
4. Follow the inputs to the system and record what happens to them to identify the activities/functions.
5. Examine the model for weakness – waste in process-based system, failure occurrences in product-based systems.
6. Determine actions to improve the existing system.

In both cases it is important to control the amount of detail through the construction of a set of top down diagrams as follows:

- Start with the construction of a simple overall model of the whole system of interest. Here Miller's[2] rule of 7±2 functions/activities is useful here as a guide

[2] George Miller in his 1956 seminal paper "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information" did what said on the tin. In simple terms all humans have a processing limit and when faced with complexity beyond this limit we make mistakes, we start to miss things.

- For each “high-level” element identified on the overall model construct a detailed lower level diagram. Again the complexity rule of 7±2 functions/activities should be used.
- If necessary consider even lower-level diagrams – 3 levels of detail is usually sufficient.

It is also very important when starting to construct a Sequence Diagram that the team understand the context of the problem, particularly that related to the lifecycle phase. For any System of Interest it is often possible to construct a variety of Sequence Diagrams for different points in time. Figure 1, for example, shows an installed domestic automatic washing machine doing its day-by-day activity; i.e. washing dirty cloths. In other words, it shows the operational phase of the system’s life cycle. It is possible to construct other lifecycle views such as installing or repairing a washing machine. The fact we can view any system from many different perspectives is important to recognise when constructing a Sequence Diagram. It is essential that the perspective is defined clearly and everybody in the team is clear what view is being taken. Some projects may demand that we capture several perspectives. In such instances, it is recommended to start with the day-to-day operation of the System of Interest and then consider the other views later.

Tips for Constructing Sequence Diagrams

- Do consider using sticky notes and white-boards for the early drafting work. It can be useful to brainstorm system functions or activities onto stick notes, arrange them on the white-board and then draw the interconnections. If you have much more than the seven small groups of functions or activities look to combine functions into “super” functions.
- The initial diagram will require several iterations and a white-board provides a convenient medium. Furthermore, it is useful if team members can “sketch” out their ideas to show other team members. If white boards are not available, flip charts are an alternative, but are less easy to modify. Software tools are available to capture the outcome, but in general they are less useful for constructing diagrams using a team.
- Consider the operational view of the system first. It is possible to create many different models of any one system (usually based on phases of the lifecycle of the system). This may well be necessary at some point, but when initiating a modelling exercise it is best practice to start with the operational view, i.e. the system has been designed and installed and consideration is aimed at its day-today operation.

- The initial drafting of a Sequence Diagram should consider every possible input and output. This often results in a very “busy” diagram and there is a tendency to either ignore inputs or outputs because they are considered not important. It is preferable to capture all these and rationalise and simplify the diagram later. Indeed, having captured all the inputs and outputs the diagram can be simplified by collecting similar inputs or outputs together and creating a collective-name that can be captured in the system dictionary. For example, the washing machine shown in Figure 1 has the input labeled “CLEANING_AGENTS ”. This is a grouped input comprising detergent, fabric softener, water softener, stain removers, bleach and or dye stuff. But do keep a dictionary to record any definitions of any inputs or outputs. For example CLEANING_AGENTS can be defined in the dictionary as

  CLEANING_AGENTS = (DETERGENT) + (FABRIC_SOFTENER) + (WATER_SOFTENER) + (STAIN_REMOVER) + (BLEACH) + (DYE_STUFF)

  Hence the definition of Cleaning_Agents above means, that Cleaning_Agents is equivalent to optionally Detergent and optionally Fabric_Sotener etc. The use of an underscore to join words in to be absolutely clear that the definition is for a “thing” and not a typo! As a general rule, it is always best to simplify where possible, but this does need a dose of pragmatism in that situations can be over-simplified.

  This uses a set of standard conventions is used to explain the make-up of inputs or outputs. These are shown in table 1:

Convention	Shorthand
IS EQUIVALENT TO	=
AND	+
EITHER-OR	[option 1, option 2]
ITERATIONS OF	{items}
OPTIONAL	(item)

Table 1: Dictionary Conventions

What Goes Wrong: The limitations of Sequence Diagrams

An Sequence Diagram is a very simple but powerful tool for exploring the boundary and “workings” of system – existing or new. It does build a model of the system of interest, but like ALL modelling methods, it has limitations. The following outline these limitations and where possible propose approaches to minimise their effect.

- Sequence Diagrams are abstract models that focus on the system’s input-output transformation. The resulting model is not a physically related model and teams, particularly inexperienced teams, try to construct a diagram that reflects the likely physical manifestation of the system.

- Sequence Diagrams do not readily lend themselves to the simultaneous capture of multiple modes of operation. Many systems have several different modes of operation (often due to dealing with different scenarios or lifecycle phases). In consequence the result is a model that is difficult to read because it attempts to mix modes of operation or a number of diagrams that individually are ephemeral.

Success Criteria

The following list represents a set of criteria that have been found to be useful when constructing a Sequence Diagram.

- Team size between five and eight.
- Team constitution covers system life cycle and potential technology.
- Use an experience independent facilitator.
- Plan for a one to two hour session.
- Draft the Sequence Diagram on a large white-board or equivalent using sticky notes for the functions or activities. Be wary of constructing the diagrams directly in software! People should be encouraged to draw out their understanding – if they are intimidated by not being able to drive the software they will agree too readily with a team member view rather than explore their view.
- Show the draft Sequence Diagram to other interested parties for verification and validation.