

The Systems Engineering Tool Box

Dr Stuart Burge

“Give us the tools and we will finish the job”

Winston Churchill

Viewpoint Analysis (VPA)

What is it and what does it do?

Viewpoint Analysis (VPA) is a tool that allows a team to identify, structure and document the requirements of a system. The outcome is a *Viewpoint Structure Chart* like the one shown in Figure 1. This chart presents a hierarchical decomposition of the system functionality that has been identified as necessary to meet the prime system’s operational requirements¹ together with the external functionality of the wider system of interest. The chart also shows how the non-functional requirements impact and constrain the system’s functional requirements.

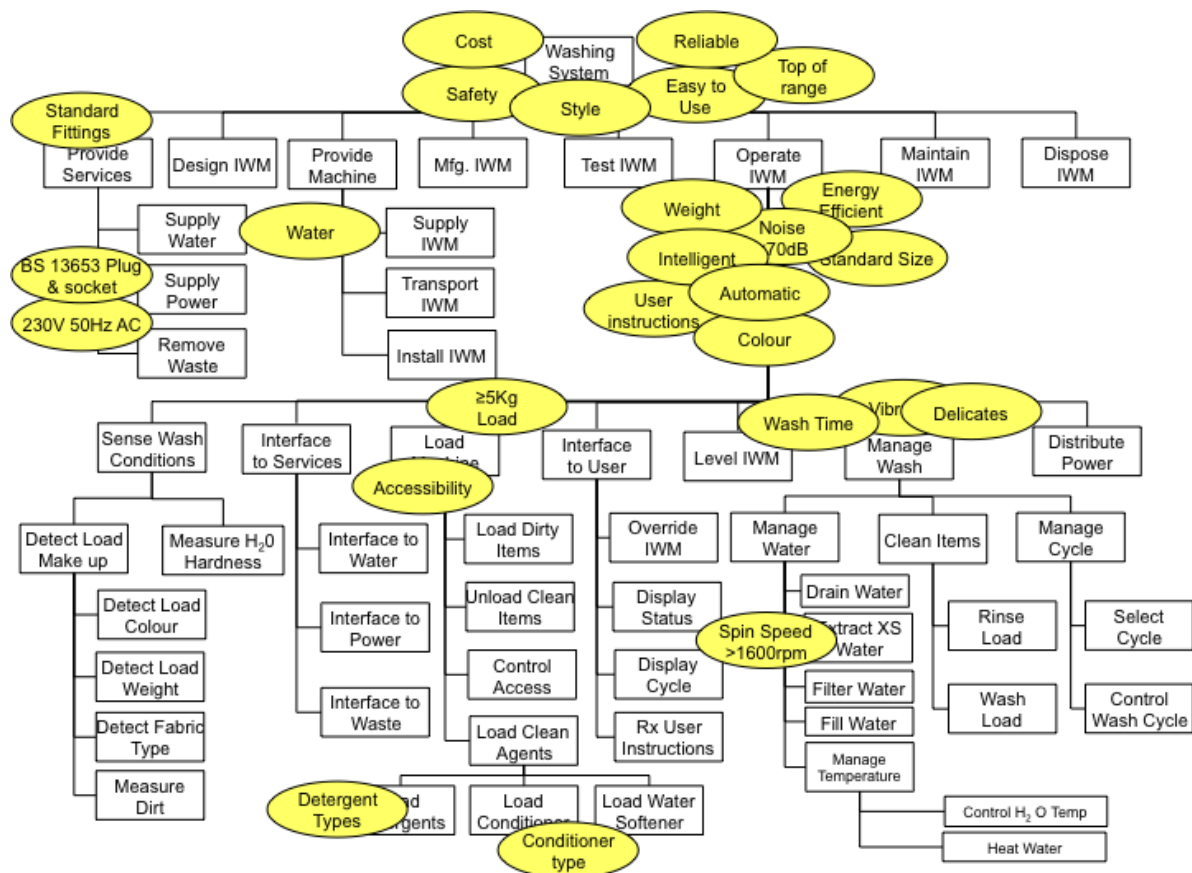


Figure 1: An example Viewpoint Structure Chart

¹ Reminder: Every system has an *Operational Requirement* that defines the major purpose of the system together with any overarching constraints.

The benefits of Viewpoint Analysis are:

- a simple tool for diagrammatically showing the hierarchy/structure of functional and non-functional requirements.
- a diagrammatic way of highlighting requirements missing from the original customer requirements.
- a process that encourages the team to consider not only the prime system but also the wider system. All systems operate in an environment; failure to pay attention to that environment will lead to failure.
- outcomes that can help to structure the concept design and project management.
- a process and outcomes that can help to gain more understanding about the prime system and what all the customers' expect.
- outcomes that can help in showing the customer that you understand what they are after.
- an approach that when used within a team context it allows the whole team to share information and agree at a common understanding.

Why do it?

There are three facts that drive us towards using Viewpoint Analysis:

- Customers never provide a complete or consistent set of system requirements
- Every system has multiple customers or stakeholders
- Given a complex situation only a partial understanding will be obtained from single point of view.

The first fact implies that is up to the System Provider to interpret the customer's expressed requirements and develop and derive a complete, consistent and unambiguous set of system requirements as the foundation for system design. It follows from the second and third facts that a more complete understanding will be obtained if several viewpoints are taken that explore the different perceptions of the system's stakeholders.

Where and when to use it?

Viewpoint Analysis is used to help understand and develop a set of requirements. It is particularly useful when we have:

- limited information from the customer such as an operational requirement or idea of a potential operational requirement. In this situation we use VPA to derive, structure and present requirements.
- too many requirements at multiple levels. In this situation we use VPA to generate a logical framework for handling large numbers of requirements.

Who does it?

VPA is team-based tool that fundamentally relies on the experience and expertise in that team. It is important to emphasise that the quality of the outcome is dependent upon the team and hence team selection is critical. The VPA team should comprise members who have good knowledge of:

- existing systems
- expected usage profile
- life cycles of similar systems.

Time size is also an important consideration and the recommendation is for four to eight members. Below four, and there is perhaps not enough critical mass to represent all the stakeholders adequately. Above eight, creates management issues, which often leads to bureaucracy and consequent slowing of progress.

There is great benefit in terms of quality of output and time efficiency if the VPA sessions are facilitated by a VPA craftsman. This is particularly important for virgin teams. VPA requires the team to identify system functionality as the basis for division and subsequent structuring. As a general comment, people, irrespective of their background, find it difficult at first to think functionally and an experienced facilitator can help the team “think” the right way.

How to do it?

The concept

Viewpoint Analysis has two distinct phases

- A divergent thinking phase to generate system requirements.
- A convergent thinking phase to structure and make sense of the output from the divergent phase.

The mechanism for the divergent phase is “brainstorming” whilst the convergent phase “divides and conquers” by taking two separate viewpoints:

- A *Functional Viewpoint* which looks at the system in terms of its functionality.
- A *Non-functional Viewpoint* which looks at the system in terms of its constraints.

The approach adopted in Viewpoint Analysis is to separate functional from non-functional requirements, develop a structure between the functional requirements and show how the non-functional requirements impact on them. In other words, the tool concentrates on developing a structure for the functional requirements of the system constrained by the non-functional requirements.

The practice

In order to conduct a Viewpoint Analysis a number of steps are undertaken as shown in Figure 2.

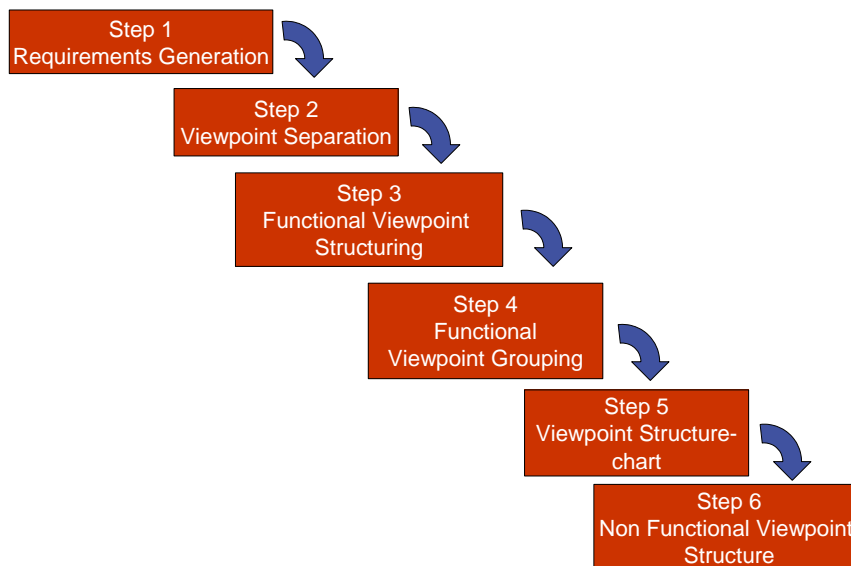


Figure 2: The Viewpoint Analysis Process

STEP 1: Requirements Generation

The first step is to determine the contents of both the functional and non-functional viewpoints (these are the system requirements). The aim is to try to write down ALL possible requirements for the prime system throughout its anticipated life.

The key point to note here is that what the customer has expressed is typically only a small part of the complete requirements set. The team is going to, and has to supplement these from their experience and through logical thought. This particular step is crucially dependent upon the composition of the team. It is where all the various requirements from different team members are collected.

The best process for undertaking the Requirements Generation Step is a straightforward brainstorming session with the requirement captured on what is called a *Viewpoint Bubble Diagram*.

To illustrate this first step and provide more practical detail a case study of an Intelligent Washing Machine will be used. The starting point for this is a set of “Customer Requirements”. In this example these have been expressed as a simple “Product Brief” as given in Appendix A

This document forms the starting point for the VPA. The team members are given time to read the document and make notes if necessary. The next step is to undertake the brainstorming session where the team members are invited to define any requirements they see fit for the Intelligent Washing Machine (the prime system) throughout its anticipated life cycle. As will become clear a little later it is very important at this point not to just focus on the prime system undertaking its prime

task. It is important to consider what will happen to the prime system throughout its life. Effectively, the team needs to identify and capture the life cycle of the prime system. For product-based systems like the Intelligent Washing Machine (IWM), this will include:

- Marketing
- Design and Development
- Production
- Testing
- Storage
- Transport
- Selling
- Installation and commissioning
- Operation
- Maintenance
- Disposal.

This is not an exhaustive list but is typical of the lifecycle activities. Figure 3 shows the *Viewpoint Bubble Diagram* for the IWM.

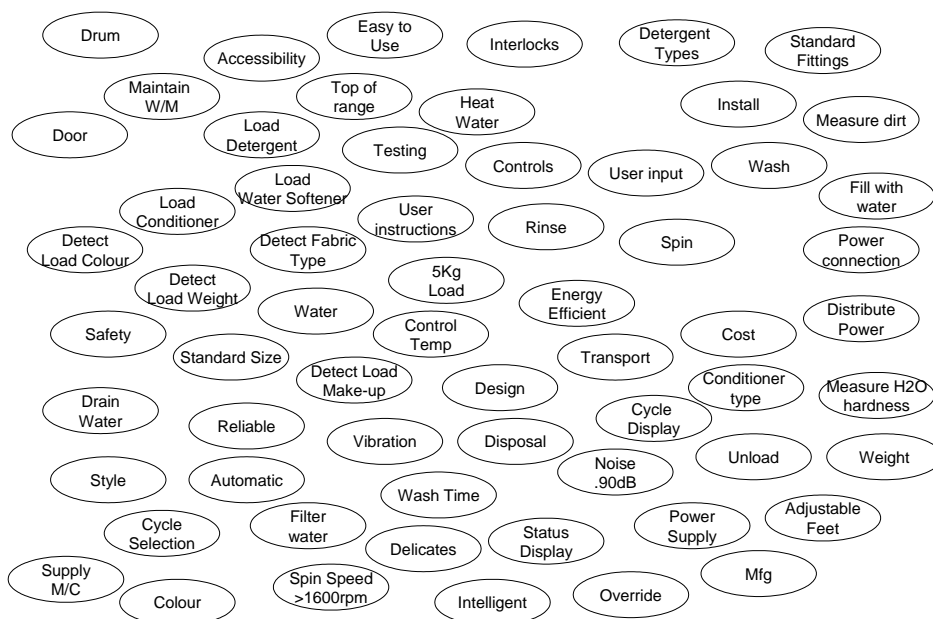


Figure 3: Viewpoint Bubble Diagram for the Intelligent Washing Machine

It should be clear from Figure 3 why it is called a “bubble” diagram; each requirement is captured in its own bubble. In practice the requirement can be captured on sticky-notes. Cross-referencing with Appendix A will show that Figure 3 contains a great deal of information not found in the original requirements document. For example, the ‘team’ has included requirements for *supply M/C* and *fill with water* which were not expressed in the original requirements document. It is these team-generated requirements, which often make up 80% + of the final requirements set, that this step is trying to extract. The knowledge of these requirements is inside the heads of the individual team members, and the purpose of this step is to extract that experience

and knowledge. Such unexpressed or un-stated customer requirements arise from one of two mechanisms;

Firstly by experience, there is somebody in the team, who in previous system development has recognised the need for a particular requirement. The life cycle aspects are typical of this type – because there is a maintenance person in the team they will through their personal experience, ensure that it is identified and captured as a requirement of the system. This mechanism has a strong influence on team selection. Ideally all of the life cycle activities need representation. However, this can lead to large teams that should be avoided. In practice, a balance is sought between team size (<8) and team coverage.

Secondly by logic, some requirements follow from the presence of others. The customer requirement document talks about using domestic water supplies, best wash cycle, etc but there is no mention of the fact that one of the requirements of the system will be to fill up with water. Sheer common sense says that the machine will have to do this – it is a “requirement”.

It is common during this brainstorming session for team members to “think” in terms of solutions and not requirements. This is perfectly normal. We live in a world full of objects. Our first step in learning a language is done through naming objects. It is hardly surprising that we are naturally inclined to express our desires as objects, especially if we already know the name of an object that does the job. For example, with the washing machine there are several instances of solutions such as:

Interlocks
Spin
Adjustable feet
Door

Each of these “objects” does something and we need to convert the solutions into logical requirements, that is, into logical functions.

<i>Object</i>	<i>Function</i>
Interlocks	Control Access
Spin	Extract Water
Adjustable feet	Level
Door	Load clothes

This conversion can be done as the brainstorm progresses or after the brainstorm has finished.

Another common situation when brainstorming requirements that we are often economic with our language and describe functions by just the verb This is not a bad thing to do, the requirements generation step is about quantity not quality. It is, however, good practice to review the brainstormed requirements and convert verb only functions into full verb noun descriptions. For example Figure 3 has captured a “bubble” containing just the verb “RINSE” – is this a function and if so what is it rinsing? In this case the verb is supplemented by the noun “load” – that is the function is RINSE LOAD.

A further common occurrence at this stage is to convert every requirement into a “verb-noun” combination. People will find ways to convert constraints (particularly those that they think are important) into what looks like a function since it is a “verb noun” description. For Example:

Provide reliability
Generate style
Increase maintainability

These, strictly speaking, satisfy the “verb noun” definition of a function, but are not really functions of the system – a system does NOT do reliability or style. When the system has been built you would not be able to point to the part that provides reliability as you would to the crumb tray as the implementation of the “collecting crumbs” function. A function has to be implemented, which can be used as a test of a requirement. The key here is to examine the noun that follows the verb - it has to be a common noun as opposed to an abstract noun².

Look for verb noun phases that describe something the System has to DO as opposed to something the system has to BE

Having generated the system requirements it is now time to make sense of all the information. This is achieved by a number of steps that will ultimately lead to a simple hierarchical model of the requirements proposed system. Each of these convergent steps simplifies by looking for natural order and structure with the system based upon the system functionality. The first of these steps is to separate the requirements into functional and non-functional requirements.

STEP 2: Viewpoint Separation

The next step in the approach is to sort the requirements into either functional or non-functional types to provide the two basic viewpoints.

The *Functional Viewpoint* is defined as:

A logical partitioning of the system into the functions that are necessary for the system to achieve its operational requirement.

The *Non-functional Viewpoint* is defined as:

Requirements that modify or constrain the system

To undertake this step is a matter of systematically examining every requirement on the viewpoint bubble diagram. Functional requirements have the following characteristics:

- A functional requirement defines what has to be done – not how it is done or how well it is done. A functional requirement is a function of the system

² Abstract nouns are things that exist but you cannot touch such as reliability, style, loyalty etc.

- There are many levels of functions in a system. We should attempt to determine all of them.
- Functional requirements are a verb or verb–noun phrase.
- Having a verb in a requirement is a necessary but not sufficient condition for a functional requirement.
- Functions often transform inputs to outputs.
- Functional requirements should be implementation independent – they specify the “job” to be done – not how it is to be done.
- When identifying functional requirements we need to be clear on what is the system of interest.

The last characteristic is quite important in ensuring a holistic view is taken. When conducting a VPA it is important to distinguish between the Prime System and the System of Interest. Let me use the Washing Machine example to illustrate the difference. The prime system is the washing machine, but this is only part of a much bigger system of interest. The system of interest therefore defines that bigger system that we NEED to consider in order to design the best possible prime system. The system of interest can therefore be defined as the prime system and those elements in that prime system environment. There is danger here in that, unless we are careful, the environment is wrongly restricted to those elements that are necessary for the IWM to wash clothes. We NEED to consider the whole life cycle. For example:

Supply IWM
Transport IWM
Install IWM
Test IWM
Design IWM
Dispose IWM
Manufacture IWM
Etc.

are all functional requirements of the “System of Interest” and describe what this system has to do. These are in fact life cycle-functions of the prime system (the IWM). Step 2 therefore requires us to identify the functional requirements of the system of interest and the prime system. If it cannot be decided whether a requirement is functional or not, it should be put in the functional viewpoint.

It should be remembered that customers tend to specify solutions and performance levels rather than functionality. We should consider every bubble to see if it implies a function - which we should deduce and record on the bubble diagram. Our aim here is identify *all possible functionality* of the proposed system of interest and prime system. Note that we can always choose not to implement functionality, but if it is not identified in the first place we will never have the opportunity to make that decision.

Figure 4 show the Viewpoint Separation for the IWM. This diagram shows the “first pass” with three groups:

- Obvious functional requirements
- Obvious non-functional requirements
- Items that could fall into either category.

It is the last category that is interesting and this often shows up our attempt to brainstorm all the requirements. Some are just poorly defined while others have a functional and non-functional derivative. It is necessary to discuss these and redefine where necessary and derive the appropriate functionality and constraints elsewhere. In this case those items that were not clear were reassigned as shown in Figure 5.

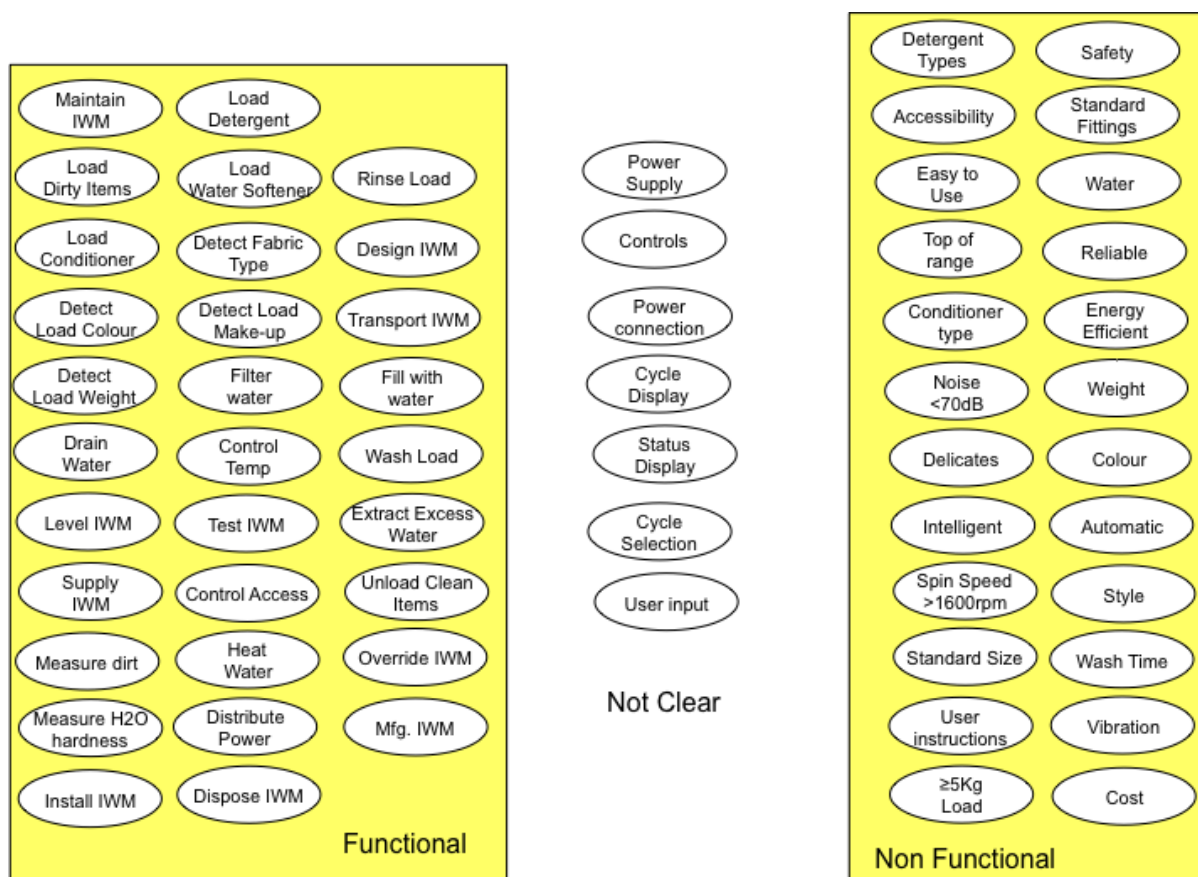


Figure 4: Separated Viewpoint Bubble Diagram

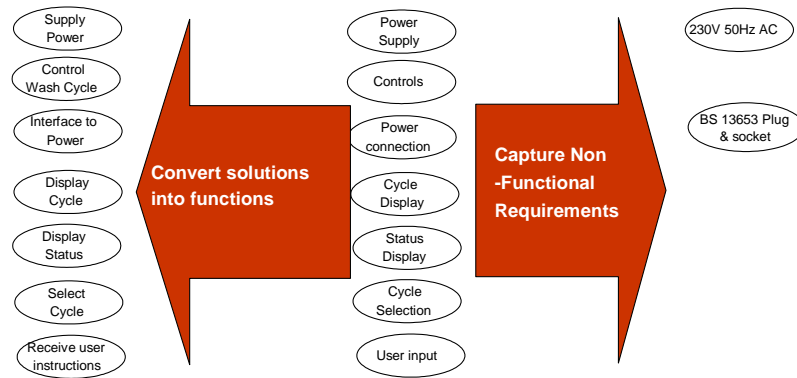


Figure 5: Reassignment of the Requirements that were not clear.

STEP 3: Functional Viewpoint Structuring

Functional viewpoint structuring is central to VPA. The aim of this structuring is the development of a logical functional requirement hierarchy. The non-functional viewpoint requirements are temporarily discarded and consideration is given to logically structuring the functional viewpoint. This is done in order to split the analysis task and thus be in control of the amount of detail. The first step in structuring the functional viewpoints is to identify what are called the *External (or Bounding)* and *Internal (or Defining)* viewpoints.

The *External (or Bounding) Viewpoint* is defined as

The External viewpoint is a view of the prime system from the outside – this highlights the functionality external (particularly life-cycle activities) to the prime system and defines the system of interest.

The *Internal (or Defining) Viewpoint* is defined as

The internal viewpoint is a view of the prime system from the inside and is used to describe the internal function of the prime system.

I hope you can see from these definitions a link with the discussion about Step 2 with regard to life cycle functions and internal functions. Figure 6 shows the first attempt at defining the contents of the Internal and External Viewpoints.

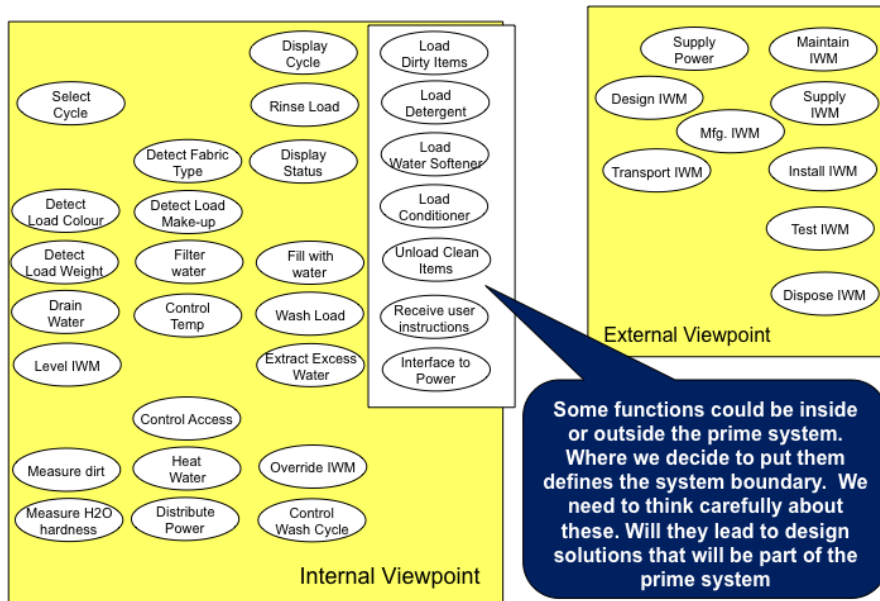


Figure 6: Identifying the Bounding and Defining Viewpoints

From Figure 6 you should note that some of the functions could fall into either category. This is typical of interface functionality and it often requires us to have functionality to describe both ends of the interface. For example there is a need in the Bounding Viewpoint to have a *Supply Power* function, but equally the Defining Viewpoint needs a function to interact with this external function. This could be written as *Receive Power* or *Interface to Power*. In this particular system these two functions will turn into physical objects that “do” the functions. This step is not easy and makes us think about the boundary of the prime system and whether we have identified all the interface functionality. Answering these questions often leads to inclusion of new but necessary functionality as shown in Figure 7.

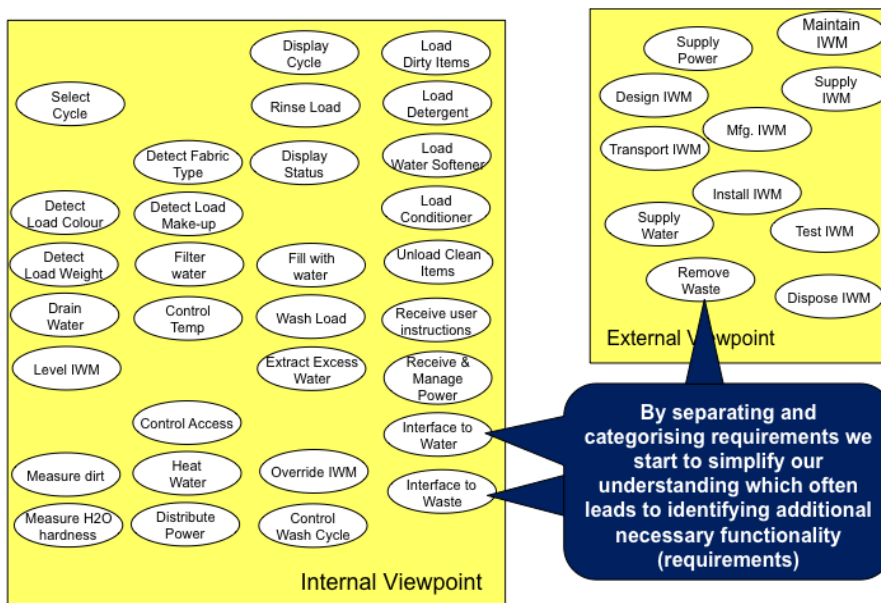


Figure 7: Updated Bounding and Defining Viewpoints

STEP 4: Functional Viewpoint Grouping

The next step is aimed at further refining the structure of the system of interest. This is achieved by forming logical groups of functions in both the Internal and External Viewpoints.

As simple rules of thumb, functions should be grouped such that:

- in any one level of grouping there is a maximum of five functions
- the groups should be named (if you have difficulty deciding on a name the grouping is not logical so look for another grouping).
- The group name should be a verb-noun description to maintain the functional description. We tend to drift back of object-oriented behaviour.

There is no other guidance and it is up to the team to decide what they think is the best grouping. This often leads to much discussion as some functions could easily reside in several of the chosen groups. At the end of the day it does not matter providing the team do agree and are comfortable with the final groupings.

Figure 8 shows the outcome from such a grouping exercise but does not show the debate that went into achieving this outcome.

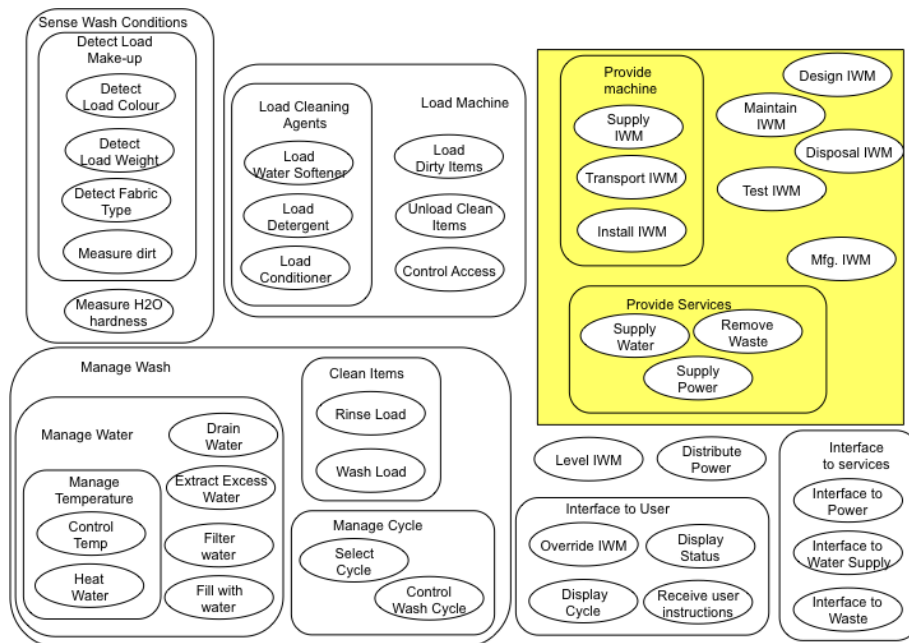


Figure 8: Grouping of Functional Viewpoints

STEP 5: Functional Viewpoint Structure Chart

The penultimate step is to convert the structured viewpoint bubble diagram into a Functional Viewpoint Structure Chart as shown in Figure 9. When forming the chart the bounding viewpoint functions are placed at the same level as the prime system. This is actually a very important aspect of the tool. There is always a tendency to focus and concentrate on operational aspects of the prime system: what the prime system has to do in order to achieve its main purpose. While this is certainly important, it is also important that the prime system is designed for:

- disposal
- maintenance
- installation
- delivery
- storage
- sales
- test
- manufacture
- etc.

These aspects can only be designed in if they are considered at the same time as the design for operation. By elevating the External Viewpoint functionality to the same level as the prime system level ensures that these “design for” factors are given due attention.

There is, however, much more to a Functional Viewpoint Structure Chart. Firstly, the Functional Viewpoint Structure Chart conveys a great deal of information in a single diagram. It provides a rather neat summary of the totality of the requirements of the prime system and the interactions with the life cycle functionality and other external functionality. This makes it easier to explain the totality of the system as well as being able to delve as necessary into the detail. It is a very powerful representation of the system and can be used to demonstrate understanding to customers and internal reviewers.

Secondly, the Functional Viewpoint Structure Chart is very similar to a Work Breakdown Structure (WBS) used by Project Managers. In fact, it is probably a better basis for a WBS since:

- it was developed by the team and therefore has considerable “buy-in”.
- the structure is functionally based and therefore is not biased towards any particular solution. It will therefore be robust against project change.

Of course, as we develop the Functional Viewpoint Structure Chart we should be constantly reviewing it for missing functionality.

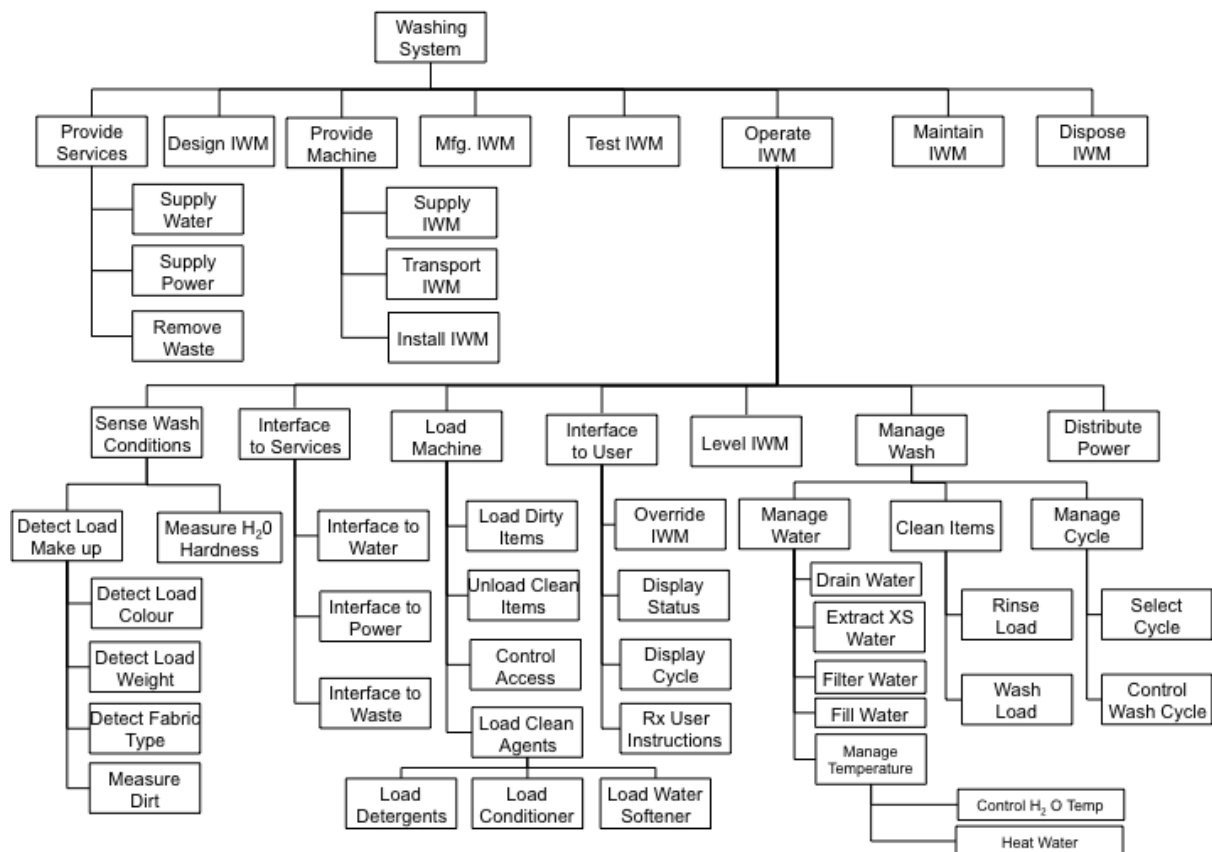


Figure 9: Functional Viewpoint Structure Chart

STEP 6: Non-Functional Viewpoint Structure

The final stage in viewpoint analysis is to structure the non-functional requirements. This can be achieved by finding the “best” place for a particular non-functional requirement to sit in the Functional Viewpoint Structure. Each of the non-functional requirements should be considered systematically and placed against an appropriate functional “box”. Care should be exercised to ensure that the non-functional constraints are placed high enough in the functional hierarchy.

This step can be very useful in identify further missing requirements – particularly non-functional performance requirements (how well a particular function has to perform) which align to the lower functional boxes. Indeed, every “box” on the Functional Viewpoint Structure Chart should have a series of non-functional constraints impacting upon it that detail how well that function has to perform. This can be seen in Figure 10 which shows the “final” Viewpoint Structure Chart for the IWM. Note here the large number of lower level functions that have no non-functional requirements impacting upon them. It is important to note that a functional box without any non-functional requirements does not mean we rush back to the customer asking for more information. Quite the opposite in fact – we need to think and decide how well a particular function has to perform. The implication of this is more work to develop a specification of what the constraints are. These are developed from the customer’s requirements document together with other sources of information such as national standards and the physics of the problem.

What Goes Wrong: The limitations of VPA

Is the Viewpoint Structure Chart Correct? It is sometimes of concern that the team feels their structure diagram is wrong. These fears arise from the fact that if five individuals are asked to determine a viewpoint structure for the same problem they will generate five different diagrams. However, these will be five correct diagrams! The differences will arise from Step 5 when forming the groups of functions. Here there are typically many choices all of which are potentially correct. It is important at this point to remember that the purpose of Viewpoint Analysis is to generate more knowledge about a proposed system. As long as the viewpoint structure represents the whole of the problem, it does not matter if it is different to another potential viewpoint structure.

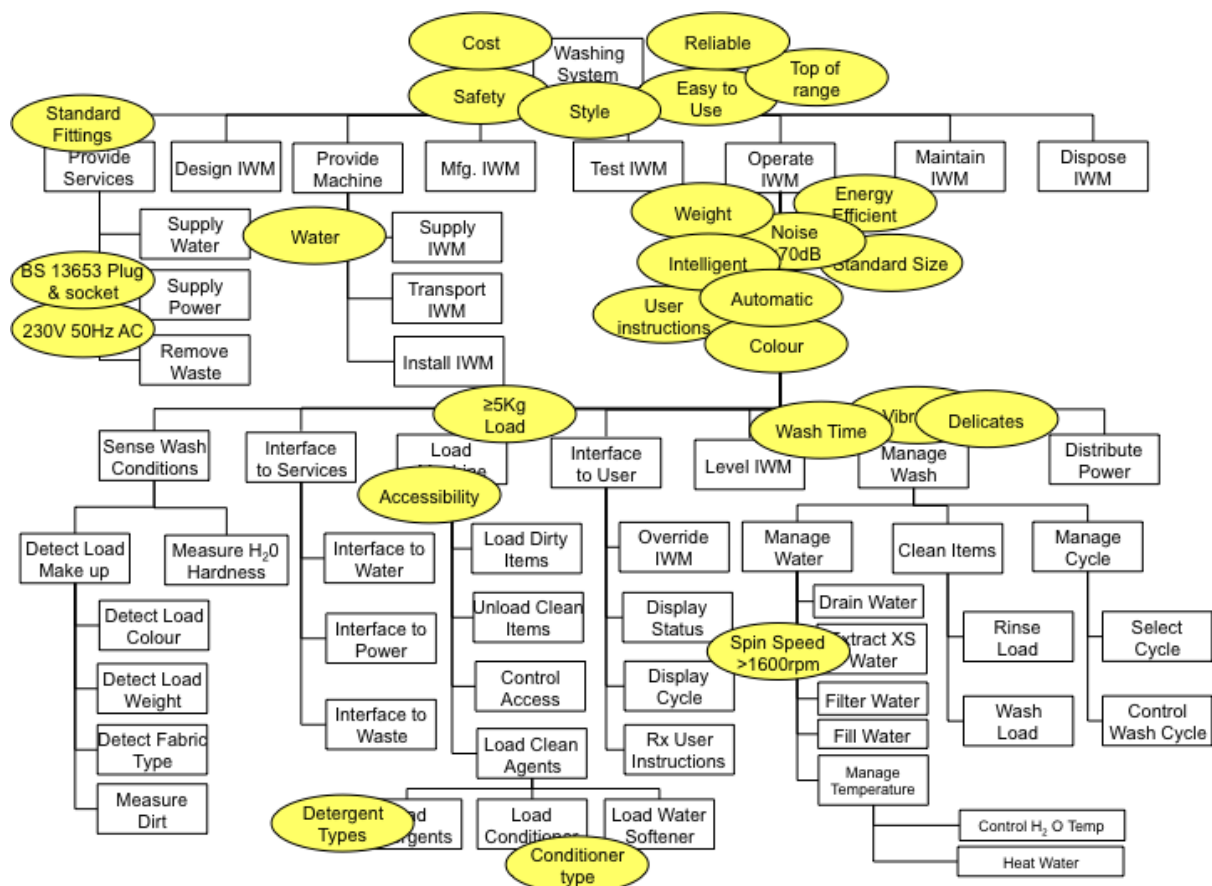


Figure 10: The final Viewpoint Structure Chart

Success Criteria

The following list represents a set of criteria that have been found to be useful when undertaking a Viewpoint Analysis. Ignore them at your peril!

- Team size between 5 and 8.
- Team constitution covers system life cycle and potential technology.
- Use an experience independent facilitator.
- Plan for one half-day's effort. Irrespective of the problem a VPA will take about ½ day to complete comprising 60 to 90 minutes brainstorming (this is the limit of human endurance for tools like brainstorming) followed by 2 hours of sorting.
- Focus on functionality and not solutions.

Appendix A: Requirement for the Intelligent Washing Machine

1.0 Background

1.1 Studies demonstrate that there are an increasing number of single persons with relatively high disposable incomes. Their lifestyle and priorities are such that they wish to minimise domestic chores. An opportunity therefore exists for an intelligent washing machine that is capable of automating much of the current manual functionality associated with washing domestic items.

1.2 The machine will compliment our existing top of the range model and there is an opportunity for the machine to be a “lifestyle statement” and it therefore must be attractive and distinctive.

1.3 The market studies indicate that there is a potential total market of 250,000 machines per annum for this sector (our share is estimated at 20% to 30%) and current spend analysis indicates a selling price in the region of £550 - £650 (including VAT).

2.0 Technical Requirements

2.1 The machine will be of standard size (595x580x850) and take a standard 5kg load.

2.2 The machine must be easy to use and will be capable of determining the load make up and fabric characteristics and thence the best cleaning cycle.

2.3 It will detect mixed loads and where necessary inform the user of extreme loads

2.4 It will continually inform the user of its current status.

2.5 The machine will use domestic water and currently available detergents.

2.6 Standard single phase 230V 50Hz electricity supplies will provide the power source.

2.7 It will operate at appropriate temperatures and wash cycles most suitable for the fabric type, which are determined by the machine.

2.8 The user will have the facility to check the wash cycle and override the machine decision.

2.9 The machine will wash, rinse and spin-dry (1600 rpm is desirable) the clothing as appropriate to load and type.

2.10 The average useful life of the machine is to be 7 years and first year failure rate is to be less than 10%

2.11 The noise level at any point in the operational cycle shall not exceed 91.5db

2.12 Vibration levels should not exceed 0.5g rmps and 3.2g peak

2.13 The energy efficiency should be grade A

3.0 Legislation

3.1 The machine must conform to UK and EU safety standards.