# Stakeholders/Customer Objectives

Based on the interviews conducted with stakeholders (users and customers, specifically), we can develop an understanding of their viewpoints. The collected comments from these interviews can be clustered to generate a list of key insights or features. These features will inform a designer of what the customer/user wants to see in their solution. It could be a guiding post throughout the design process ensuring that design decisions are user centric. But we need to figure out how to utilize these features in design process as they could be all over the map. We would start by filtering the list into two classes:

* statements that specify tasks that solution should perform or specific constraints on how a task should or should not be performed. These will be catered in the functional requirements later.
* statements indicating a qualitative attribute of the solution. These are again how the solution will be evaluated, but admit a range of possibilities as opposed to a binary choice. These will be termed as objectives.

*In this document, you’re to include the following in this section:*

1. *A table describing the objective, its ideal metric, approximate metric, and data collection method. (See Table 3.6 in 22-Systems Thinking IV.pdf)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Objective*** | ***Questions*** | ***Ideal Metric*** | ***Approximate Metric*** | ***Data Collection Method*** |
| *Statement of the objective* | *What questions*  *will you ask to determine whether the objective has been achieved?* | *What quantitative measure will you use to ideally answer these questions?* | *What quantitative measure will be a good tradeoff between accuracy of data and effort required for collection?* | *How will you collect data to apply your metric, and consequently answer question?* |
|  |  |  |  |  |

1. *A brief description of chosen data collection methods, e.g. for a questionnaire you can describe the nature of questions on it.*
2. *A table describing the relative priorities and ranking of different objectives.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Objective*** | ***Relative Priority*** | ***Rank*** | ***Stakeholder*** |
| *Statement of objective* | *A number between 0 and 1 indicating priority of this objective in relation to others. Entries in this column should sum to 1.* | *Rank assigned to objective based on relative priority.* | *State the stakeholders for whom this objective is a priority.* |
|  |  |  |  |

*The following rubrics can be used to assess your work in this section:*

|  |  |
| --- | --- |
| ***Objective Goals*** | ***Checkmark*** |
| *Relevant objectives from the interview insights are included in the model.* |  |
| *Each identified objective is linked to at least one stakeholder.* |  |
| *Each stakeholder that is important for your problem is mapped to at least one objective.* |  |
| *Each chosen metric is measurable.* |  |
| *Appropriate data collection method has been thought for each objective.* |  |
| ***Qualitative Goals*** |  |
| *The solution would be satisfactory if it had only these objectives.* |  |
| *Chosen metric for each objective offers a good trade-off between accuracy of data and ease of data collection.* |  |

# Interactions View

This view is the black-box model of the system and the focus is on the entities interacting with the system through inputs and outputs. This view allows us to have a thorough understanding of the system’s boundary, which differentiates what needs to be designed by us and what is not part of our solution. The interactions view is based on identifying the *actors*, the *inputs and outputs*, and *interactions*.

## Actor

* Any physical entity that interacts with the system
* Described by **nouns**.
* Generally, provide and/or accept energy, materials, or information from the system
* Stakeholders can be actors, but actors can be objects or phenomenon as well.

## Interactions

* Interactions are descriptions of the exchange of material, information, or energy between the actors and system
* While I/O describe what is being exchanged and the actors describe who is exchanging it, the interactions describe how is it being done
* Stated as **verbs followed by nouns.**
* At this stage, internal details of the system aren’t available and so all interactions are between actors and system, or between different actors.

## Interactions Model

The interactions model can be described by stating all interactions in the form of a table and a context matrix.

*In this document, you’re to include the following in this section:*

1. *A table providing a description of all the actors related to your system.*

|  |  |
| --- | --- |
| ***Actor*** | ***Definition*** |
| *Short name of the actor. Should be a noun.* | *Longer description of who the actor is and in what setting do they interact with the solution.* |
|  |  |

1. *A table describing the interactions and their relationship to objectives from the previous section.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Interactions*** | ***Description*** | ***Actors*** | ***Objectives*** |
| *Short phrase describing the interaction. It should be written as a verb followed by noun, in terms* | *A longer description of the interaction.* | *State all the actors involved in this interaction.* | *State objectives from the previous section that are related to this interaction. Every objective demanded of a solution can only be evident to* |
| *of an actor taking an action.* |  |  | *users/customers through some interaction.* |

1. *A “Context Matrix” table, providing a visual representation of the previous table. (See Table 2.1 19-Systems Thinking.pdf)*

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Actor 1*** | ***System*** | ***Actor 2*** |
| ***Actor 1*** |  | *Actor 1 acting on the system* | *Actor 1 executing interaction on Actor 2* |
| ***System*** |  |  |  |
| ***Actor 2*** | *System carrying out an action on Actor 1* |  |  |

*The following rubric can be used to evaluate your interactions model. Table: Rubrics to assess the interactions model (Adapted from: [1])*

|  |  |
| --- | --- |
| ***Objective Goals*** | ***Checkmark*** |
| *Every actor is named as a noun* |  |
| *Relevant stakeholders are included as actors* |  |
| *Any object that the system interacts with is included as an actor* |  |
| *Every interaction is named as a verb followed by noun* |  |
| *Every actor is included in at least one interaction. If not, the actor is not relevant to problem and can be removed from the list.* |  |
| *Every interaction includes one or more actors. At the moment, the focus is on interaction with external entities.* |  |
| *Every objective from previous section is addressed by at least one interaction* |  |
| *Each interaction can be seen as a transfer of mass, energy, or information.* |  |
| ***Qualitative Goals*** |  |
| *The interactions are sufficient to fully cover the customer/user requirements.* |  |

# Use Cases

A use case is a situation in which a user would want to use your proposed system. The primary function of a solution is an obvious use case, but the goal here is to identify all possible use cases. It is important to consider secondary use cases, as well. These are ways in which the system can be abused, or the ways in which it can fail. Secondary use cases are especially important as it is here that you think about issues of safety and reliability. Operational description templates should be correspondingly developed for the use cases, but in order to save effort this can be done only for the high-priority use cases. High priority should be assigned to the use cases that are related to primary functions of the solution and those that are not completely understood and will benefit from a decomposition.

## Operational Description Template

* It should include the initial and ending conditions.
* The operator behaviors are included in the column to the left of the system.
* The columns for all other actors are to the right of system columns.
* The system can have as many subsystems as you want.
* Often times, the most difficult part is deciding the level of detail to which a function should be decomposed. The following tips can help with that decision.
  + Can you imagine a way to physically implement the function? If not, then it needs to be decomposed further.
  + Does the name of the functional block uniquely identify each behavior performed by the block? If not, then it needs to be decomposed further.
  + What if you don’t know enough about function to decompose it? It is effective to know how to decompose a function a level further than is needed for implementation. If that knowledge is missing, you’ll need to acquire it.
  + This will be an iterative process.
* The template includes interface rows (differently colored) to capture the means by which subsystems interact with each other and with actors.
* The interface row should indicate the kind of interaction (Energy transfer, Mass transfer, Information) and provide names of all inputs or outputs.

*In this document, you’re to include the following in this section:*

1. *A table providing a listing of all use cases (primary and secondary), and their allotted priority. (See Table 2.15 20-Systems Thinking II.pdf)*

|  |  |  |
| --- | --- | --- |
| ***Use Case*** | ***Priority*** | ***Interaction*** |
| *A brief description of the use case.* | *Allocated priority (High, Medium, or Low)* | *State the interaction that the use case is linked to.* |
|  |  |  |

1. *A developed operational description template for each identified high-priority use case. These are to be included as an appendix, so as not to clutter the document. (See Tables 6.5, 6.6, 6.7 in 21-Systems Thinking III.pdf)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Name of the use case goes here*** | | | | |
| ***Initial Conditions*** | | | | |
| *Provide a description of the state of the system and environment at beginning of the use case.* | | | | |
|  | ***System Name*** | | |  |
| ***Operator***  *Provide the name of the actor generating stimulus for this use case.* | ***Subsystem 1*** *The template should have a separate column*  *for each subsystem.* | ***Subsystem 2*** *Number of subsystems is arbitrary.* | ***Subsystem 3*** | ***Other actors*** *Add columns to the right of system for each actor involved in the use case, in addition to operator. These would typically be objects.* |
|  |  |  |  |  |
| *Each entry in this column describes a behavior of the operator related to the use case. The entries are listed in a causal order.* | *Each entry in this column describes a function performed by subsystem 1. The entries are listed in a causal order.* |  |  | *Each entry in this column describes a behavior of the actor, related to this use case. The entries are listed in a causal order.* |
| *Interface Row. The format of entry is:*    *Kind of interaction*  *(I/O names)* |  |  |  |  |
|  |  |  |  |  |
| ***Ending Condition*** | | | | |
| *Provide a description of state of system and environment at the end of this use case.* | | | | |

*The following rubric can be used to evaluate your interactions model.*

*Table: Rubrics to assess the use cases*

|  |  |
| --- | --- |
| ***Objective Goals*** | ***Checkmark*** |
| *Every interaction has been considered as a use case. If not, is there a good reason why it cannot be a use case?* |  |
| *Every stated use case is linked to an interaction.* |  |
| *Secondary use cases have been included in the list.* |  |
| *An ODT has been developed for each high-priority use case, including secondary use cases.* |  |
| *An initial condition has been stated for each use case.* |  |
| *An ending condition has been stated for each use case.* |  |
| *Operator column is to left of system* |  |
| *Columns of all other actors are to the right of system columns.* |  |
| *Causality has been maintained in the listing of the behaviors/ functions.* |  |
| *Interface rows are differently colored.* |  |
| *Kind of interaction has been identified at each interface.* |  |
| *At least one input or output signal has been identified at each interface.* |  |
| *Every I/O is stated as a noun.* |  |
| ***Qualitative Goals*** |  |
| *Assigned priorities ensure that all important cases are developed.* |  |
| *Decomposition into functions has been carried out to an appropriate level, i.e. every function can be easily seen as implementable.* |  |
| *All the interfaces and I/Os have been correctly identified.* |  |
| *Use cases are sufficient to cover all the customer/user requirements.* |  |

# Functional Architecture

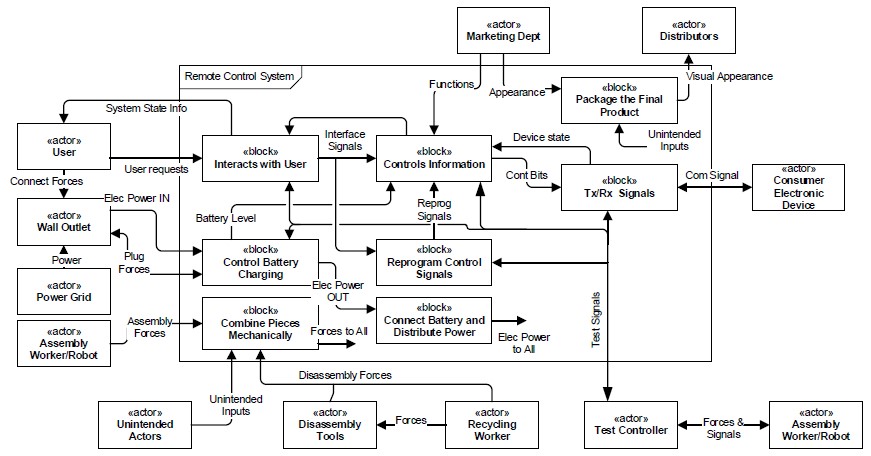
This view is the white-box model of the system and the focus is on what is inside the system. How does the system implement all the interactions? We already have a description of the system in terms of its inputs and outputs (Interactions View); it’s now time to answer how the inputs get transformed to outputs. Constructing the functional architecture requires decomposing the system into *functions* and describing those functions. We have already carried out this decomposition while developing the behavioural models of use cases.

In the functional architecture of the system, we’ll collect related functions into subsystems. The expanded view of systems involving all functions will be the physical architecture. In this context, a subsystem may be seen as a super-function and has the same characteristics as any other function.

## Function or Super-function

* A transformation of one or more inputs into one or more outputs.
* Functions may act on I/Os on system boundary, but they may also interact with other functions inside system.
* Described by **verbs followed by** **nouns**.
* They are described independent of physical form. You’re only constructing a mapping between the input signal and output signal, but not the exact procedure for such a transformation.
* Each function must be part of an interaction.

The functional architecture can be visualized as a block diagram of the kind shown in the figure below.



In order to avoid the hassle of creating such a diagram, we’ll capture the functional architecture with the help of tables.

*In this document, you’re to include the following in this section:*

1. *A table providing a description of subsystems, list of its constituent functions, I/Os, and linked interactions.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Subsystem or Superfunction name*** | ***Description*** | ***Constituent functions*** | ***Inputs/Outputs*** | ***Interaction(s)*** |
| *Name should be a verb followed by noun.* | *A brief description detailing the purpose of this subsystem (tasks to be performed).* | *A list of functions that make up this subsystem.* | *A list of inputs and outputs at the subsystem boundary. A brief description of the I/Os. You don’t need to include internal I/Os.* | *State the interaction or multiple interactions that this subsystem is linked to.* |
|  |  |  |  |  |

1. *A table that provides a visual representation of the functional architecture. This table identifies system boundary, subsystem boundaries, and inputs/outputs of each subsystem. (See Table 6.23 in 21-Systems Thinking III.pdf).*

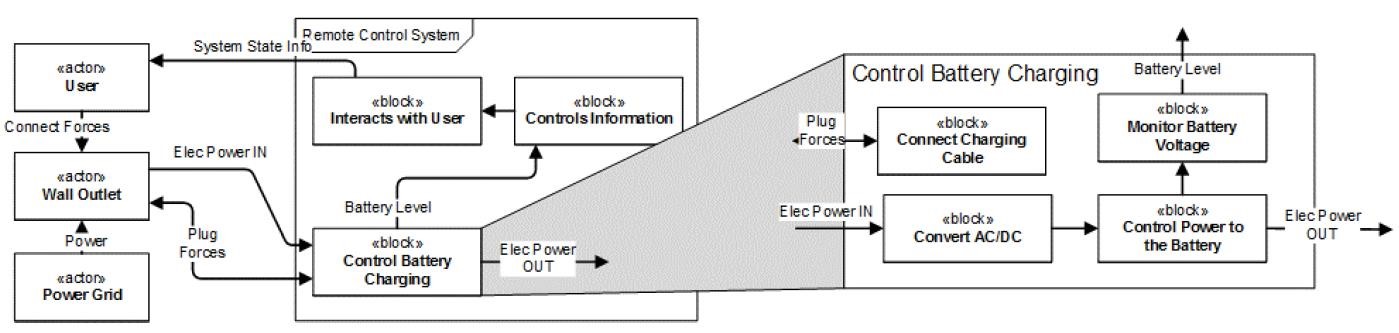
*The following rubric can be used to evaluate your functional architecture:*

*Table 2: Rubrics to assess the functional architecture (Adapted from: [1])*

|  |  |
| --- | --- |
| ***Objective Goals*** | ***Checkmark*** |
| *Functional Architecture Diagram* |  |
| *Every I/O at the system boundary goes to one of subsystems* |  |
| *Each function block (subsystem) has at least one I/O* |  |
| *Each function is related to an interaction* |  |
| *All the use cases (any priority) have been incorporated in the functional architecture* |  |
| *Each I/O and superfunction has definition* |  |
| *Each superfunction is labeled as a verb followed by noun* |  |
| *Each actor and I/O is labeled as a noun* |  |
| *Any function included in system boundary is within design scope of project.* |  |
| ***Qualitative Goals*** |  |
| *The functional architecture is sufficient to fully cover all the use cases.* |  |

# Physical Architecture

Functional architecture provides a high-level picture for the system. However, eventually every system is to be implemented, and a physical architecture will be needed. The functional architecture does not provide us with the necessary details to implement the system, and it may be required to first decompose the high-level diagram into more basic functions, also including more details in each of the I/Os. We have already carried out this task during the development of ODTs. The physical architecture is again a block diagram, and can be composed of schematics, mechanical diagrams, simplified blocks that represent small sub-circuits or other hardware, flow diagrams, UML models, or software code blocks. No matter what shape an individual block in physical architecture takes, it should be such that it can be physically implemented. The figure below gives an idea of the relationship between the functional architecture and physical one.



Since, we have already developed the ODTs, we’ll simply create a table of function descriptions.

*In this document, you’re to include the following in this section:*

1. *A table providing a description of all the functions in the system, I/Os, and linked interactions.*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Function name*** | ***Description*** | ***Inputs/Outputs*** | ***Interaction(s)*** |
| *Name should be a verb followed by noun.* | *A brief description detailing the purpose of this subsystem (tasks to be performed).* | *A list of inputs and outputs at the function boundary. A brief description of the I/Os.* | *State the interaction or multiple interactions that this function is linked to.* |
|  |  |  |  |

*The following rubric can be used to evaluate your physical architecture:*

*Table 2: Rubrics to assess the functional architecture (Adapted from: [1])*

|  |  |
| --- | --- |
| ***Objective Goals*** | ***Checkmark*** |
| *Each function has at least one I/O* |  |
| *Each function is related to an interaction* |  |
| *Each interaction has at least one function linked to it.* |  |
| *All the use cases (any priority) have been incorporated in the physical architecture* |  |
| *Each I/O and function has definition* |  |
| *Each function is labeled as a verb followed by noun* |  |
| *Each actor and I/O is labeled as a noun* |  |
| *Any function included inside system boundary is within design scope of project.* |  |
| ***Qualitative Goals*** |  |
| *The physical architecture is sufficient to fully cover all the use cases.* |  |
| *Decomposition into functions has been carried out to an appropriate level, i.e. every function can be easily seen as implementable.* |  |
| *All the I/Os have been correctly identified.* |  |

# Technical requirements (Functional requirements + Technical specifications)

The requirements provided by stakeholders are not expressed in technical terms and cannot be used by engineers to design the system. By creating precise statements expressing technical need in an objective measurable way, an engineer can devise possible solutions to the need. A list of technical requirements is the final document that can be shared among the various engineers and can serve as starting point for the engineering.

The technical requirements can be established from the functional requirements and technical constraints on the system, perhaps because of established technical performance measures. The technical requirements will be stated in the form of a table.

* Each technical requirement will be expressed as a “shall” statement.
* Each row will state a functional requirement followed by sub-items that state corresponding technical specifications.
* Possible verification strategies could be inspection, instrument test, analysis and simulation, or demonstration.

*In this document, you’re to include the following in this section:*

1. *A table with the following details.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Code*** | ***Requirement*** | ***Actors/Blocks*** | ***Inputs/Outputs*** | ***Function*** | ***Verified by*** |
| *Simple code for easy reference* | *A shall statement for the requirement.*   * *Specification 1* * *Specification 2* | *List of actors or other functions involved in requirement.* | *A list of inputs and outputs involved. A brief description of the I/Os, if not described before.* | *State the function that this requirement is linked to.* | *Provide a method that you would use to verify specifications.* |
| *P.1* | *The wall outlet shall provide power to system.*  • *Voltage*  *should be*  *220VRMS.* | *Wall outlet*    *System* | *Electrical Power In* | *Recharge battery* | *Instrument test.* |

*The following rubric can be used to evaluate your technical requirements:*

*Table 2: Rubrics to assess the technical requirements (Adapted from: [1])*

|  |  |
| --- | --- |
| ***Objective Goals*** | ***Checkmark*** |
| *Each requirement contains at least one I/O* |  |
| *Each function is described by at least one requirement* |  |
| *Each requirement has a measurable value (a technical specification)* |  |
| *Each requirement has a verification method* |  |
| ***Qualitative Goals*** |  |
| *The technical requirements are sufficient to fully cover the customer/user requirements.* |  |
| *Each requirement is achievable* |  |

# Concepts

*In this document, you’re to include the following in this section:*

1. *A morphological chart highlighting the concept fragments for different functions in your problem. (See* 26-Systems Engineering V.pdf*)*
2. *A list of possible concepts obtained through different combinations of entries in the morphological chart. Also, comment on rejected possibilities.*
3. An evaluation table comparing the shortlisted concepts and assigning them scores based on user objectives. (See Table 5.10 26-Systems Engineering V.pdf).