

# Chapter 3

## DESIGN CONCEPT

---

*In this chapter, the conceptual design phase of the project is presented. The specific software engineering approach used for the development of the application is discussed. For proper concept explanation, a step-wise approach is then taken to describe the application first from a high-level vantage point by looking at the functional architecture. We then continue by looking at the individual entity data-flows, and use the entity blocks to finally describe how the specific required theoretical techniques are to be implemented in the application environment.*

---

### 3.1 SYSTEM OVERVIEW

In order to gain a better understanding of how the system works, we need to look at a high-level description of the application that is to be developed. For proper description, two distinct approaches are taken. Firstly, we look at a design philosophy of the application, or a basic overview of how the software package's internal functional units are to be arranged, and where they fit in the overall functional picture. Secondly, these main blocks are broken down into minor blocks where we can then describe the flow of data between them.

### 3.1.1 Functional Architecture

When considering the main entities upon which the application is based, we look at four distinct groups. *Figure 3.1* shows the basic interaction and placement of these main blocks.

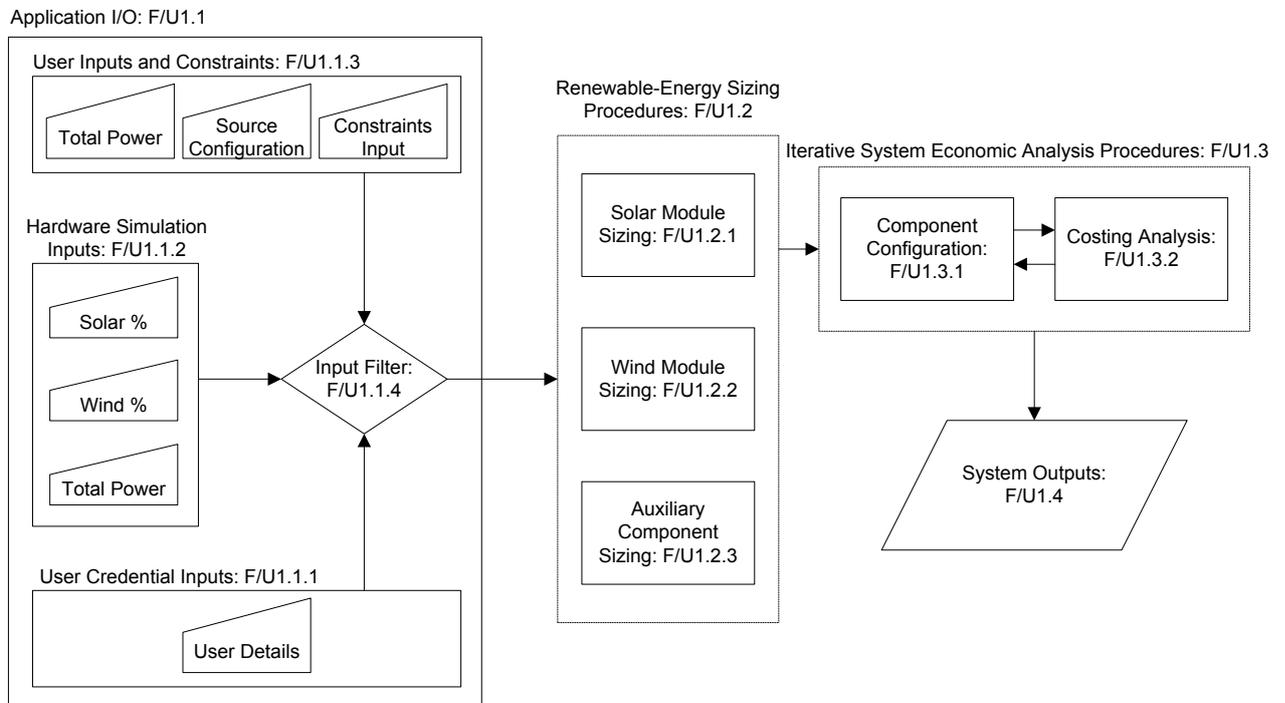


Figure 3.1 – Application functional architecture

#### 3.1.1.1 Application I/O: F/U1.1

The first group is concerned with the administrative part of the application's functionality. This includes user credential inputs and stored credential retrieval, as well as technical system inputs. The technical inputs may come from two sources: the user can choose to enter relevant and required information on a manual basis via provided interfaces, or he can choose to import data from the TSM as defined in *Chapter 1*.

#### 3.1.1.2 Renewable-energy sizing procedures: F/U1.2

The sizing procedures of the relevant renewable energy power sources are implemented in this block. This is the first main area where literary techniques are to be directly implemented. There must be noted that this block is not responsible for the technical determination of how the spread between renewable energy technologies for the REHS

should be. This information is provided to the application through either the TSM inputs, or manual user inputs as described in *section 3.1.1.1*. Using the appropriated information, this block sizes the respective solar and/or wind arrays in such a way that the required REHS energy specifications can be met. Auxiliary components required for the proper functioning and systems-integration of the renewable energy sources are also configured by this block, using appropriate techniques which follow from the literature.

### **3.1.1.3 Iterative System Economic Analysis Procedures: F/U1.3**

#### *– Component configuration*

This block serves to combine the results of the sizing procedures with the rest of the required constituents of the REHS. These required components must be added manually by the user. The application will inform the user of which components are necessary for correct functioning, but the user may add to the configuration, as long as the bare-minimum components are added to the selection. Furthermore, due to the large amount of components available on the market for each of the required parts of the REHS, a literary technique must be implemented to create functional permutations of plant configurations, which may then be optimised.

#### *– Costing analysis*

The different functional configuration permutations which are outputted by the block as described by the component configuration are scrutinized in terms of their costs. The technique applied by this block should force the application to choose the most cost-effective solution.

### **3.1.1.4 System outputs: F/U1.4**

The system output block is another administrative block, which takes the results gathered from all previous blocks, and outputs the information and chosen component configuration to the user in a professional document. System configurations are also stored using the user credentials as reference index.

## 3.2 FUNCTIONAL FLOW & BLOCK DEFINITION

To further expand on the functional architecture as described in *section 3.1.1*, we now break the main functional units down into their constituents. These sub-blocks detail the way information should flow in the application, and gives clarity as to how the system uses the information presented to it.

### 3.2.1 *Application I/O: F/U1.1*

The application I/O block is divided into four sub-blocks. These blocks, and their interaction with each other, are illustrated in *Figure 3.2*. When the application is executed, the user is presented with F/U1.1.1. Here, user credentials may be inputted which are consequently handled by the application as described in *Figure 3.2*. Once the credential information has been processed, the application continues to either F/U1.1.2 or F/U1.1.3, depending on the type of continue-flag which is enabled.

When an existing user has logged on, the ENABLE UI\_CONTINUE\_EX flag is enabled, and the application jumps to F/U1.1.2 where required system configuration information is inputted from the TSM. In the case where a new user wants to use the system, the ENABLE UI\_CONTINUE\_NEW flag is enabled, which continues as with the previous case, just without automatically importing the required system configuration data, as a TSM model should not exist in this case.

The final block, F/U1.1.4, executes when either F/U1.1.2 or F/U1.1.3 has completed execution. This block then checks if all the required system configuration information has been inputted. If this is the case, the next main block, F/U1.2 is triggered, which contain the sizing procedures.

### 3.2.2 *Renewable-energy sizing procedures: F/U1.2*

The block containing the renewable-energy sizing procedures can be subdivided into three sub-blocks as shown in *Figure 3.1*. It is important to note that there is no specific order in which these sub-blocks are to be executed, as the sub-block procedures are independent of resulting data from the other blocks within F/U1.2. This is one of the main research-applicable blocks that is to be integrated into the system.

Application I/O: F/U1.1

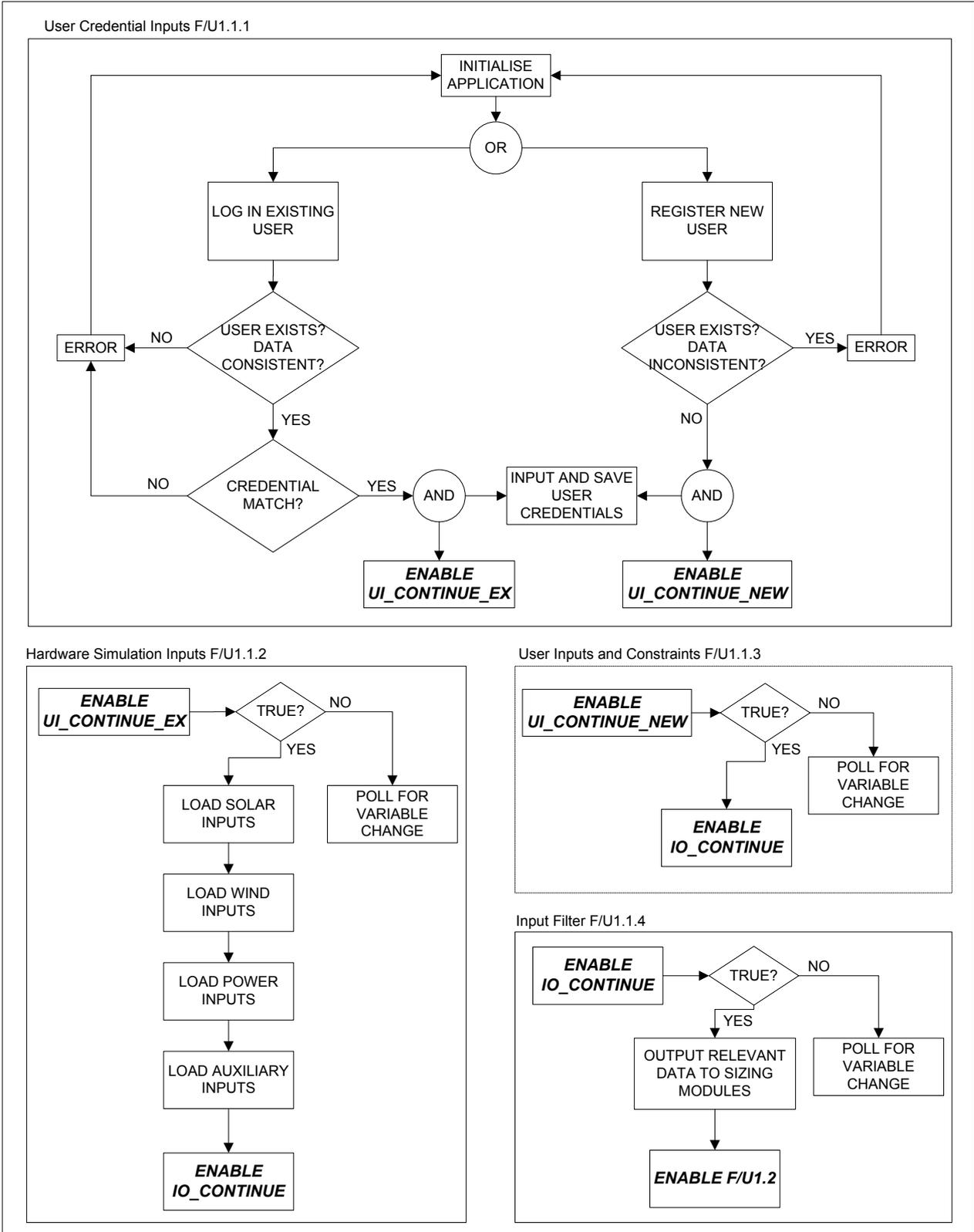


Figure 3.2 – Functional Flow Diagram for F/U1.1

Using *Figure 3.3* as illustration, we continue the data flow from the previous segment of the application as discussed in *section 3.2.1*. Two of the modules in this main block are semi-dependant on the TSM results or manual user inputs, namely F/U1.2.1 and F/U1.2.2.

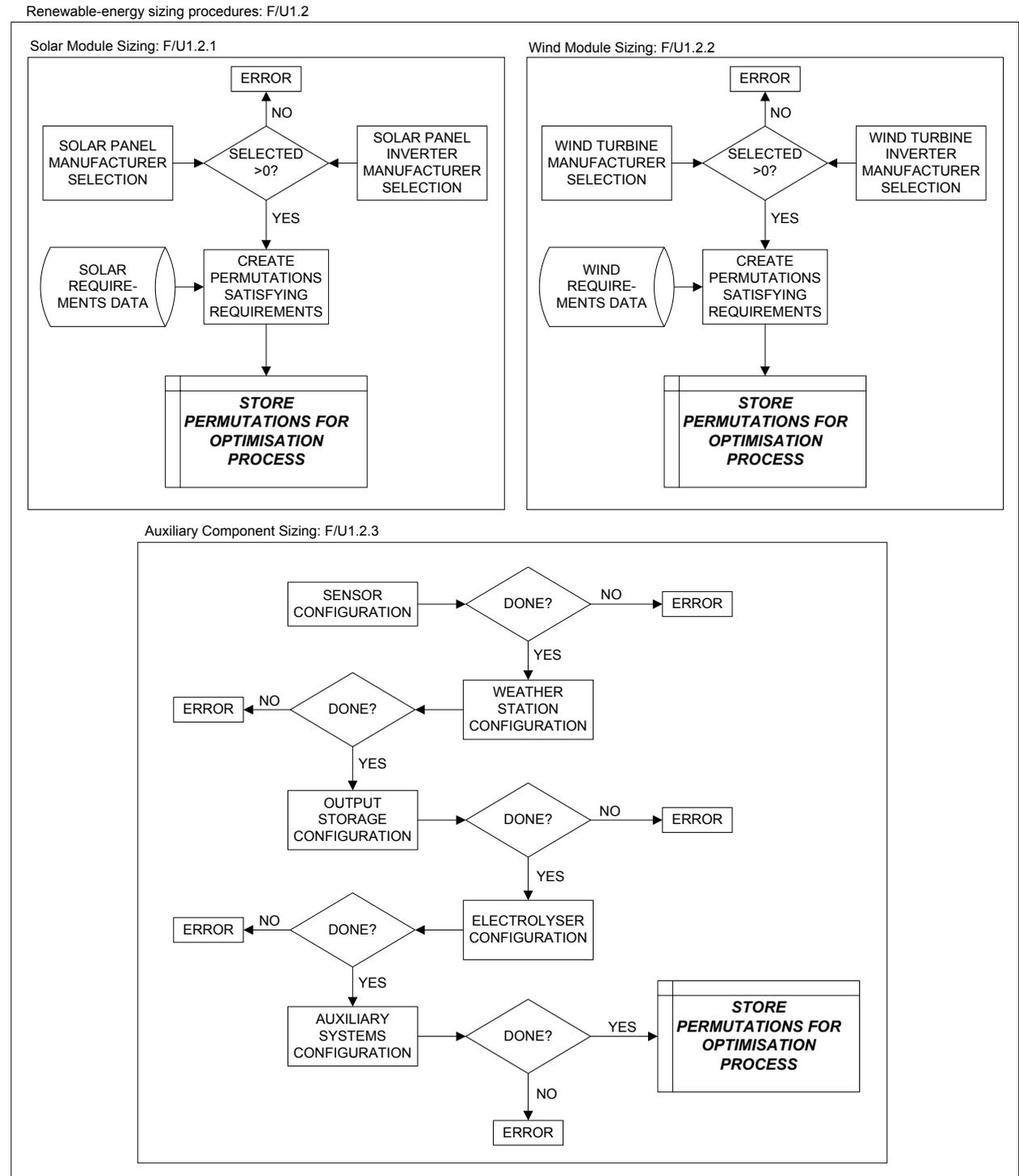


Figure 3.3 – Functional Flow Diagram for F/U1.2

As shown in *Figure 3.3*, these sub-blocks are identical in functionality; therefore to discuss a general procedure would suffice. Upon completion of the first segment of the application, the

user must now choose which manufacturers for the renewable-energy sources he wants to include in the component configuration procedure. When the manufacturers have been specified, a sizing algorithm will create permutations of viable solutions for each model of renewable energy component that each manufacturer produces, using the solar and wind power input information as retrieved by F/U1.1. Once all configuration permutations have been determined, the results are stored in an appropriate data file for use by proceeding blocks.

The procedure for F/U1.2.3 is not in itself dependant on information from the TSM. As specified in *section 3.2.1*, the auxiliary component information may be either imported from the TSM, or manually added by the user. This sub-block merely checks that all required information is present before it allows the application to continue to proceeding optimisation blocks.

### **3.2.3 Iterative System Economic Analysis Procedures: F/U1.3**

The iterative system economic analysis procedures block, is the other main research-applicable block added to the application. The main block is subdivided between two interlinked modules, F/U1.3.1 and F/U1.3.2, illustrated in *Figure 3.4*.

The optimisation techniques that are to be implemented for the system configuration process are iterative of nature. Firstly, the component data must be converted to a format which can be inputted to an optimisation algorithm. This is the purpose of F/U1.3.1. When the information has been processed, it is stored for use by F/U1.3.2. This sub-block uses an algorithm based on the genetic algorithm theory to size all possible configurations for all components, and as the algorithm executes, the solutions converge to the most cost-effective solution. The implementation of this reduces the amount of required code and ensures effective execution, regardless of component database size.

Iterative System Economic Analysis Procedures: F/U1.3

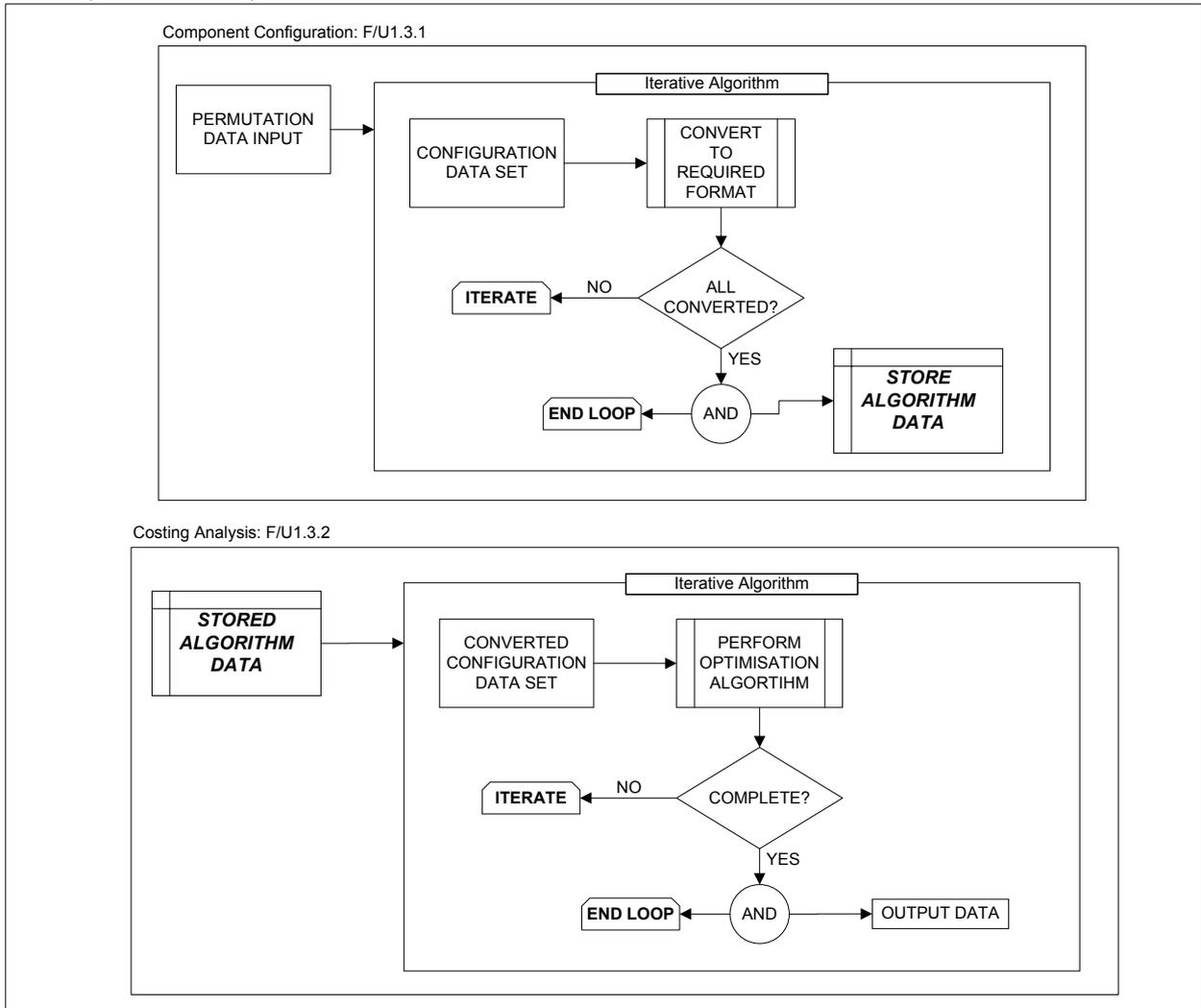


Figure 3.4 – Functional Flow Diagram for F/U1.3

### 3.2.4 System Outputs: F/U1.4

The final part of the application is created to present the client with a breakdown of the optimally sized system as determined by the TSM and ESM. Using built-in LabVIEW functionality, a professional document can be created using all relevant plant information. The basic process for the document generation process is shown in *Figure 3.5*.

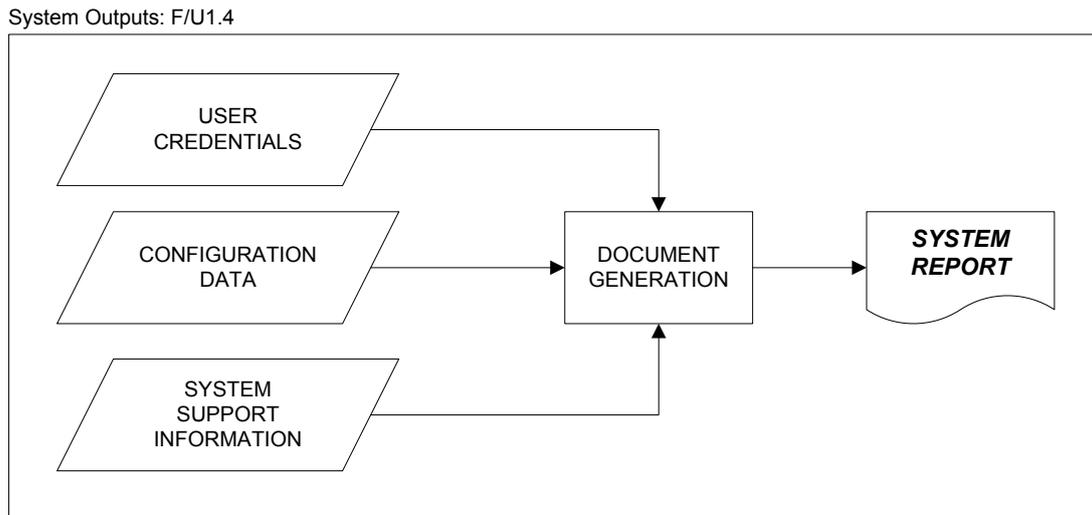


Figure 3.5 – Functional Flow Diagram for F/U1.4

### 3.3 REVIEW

To summarise, this chapter provided the framework according to which the application for the ESM is to be developed. Following a systematic approach, definite blocks have been defined for software segments, which are to be populated in the development process presented in *Chapter 4*.