Introduction

Functional Decomposition is used to break a complex system down into simpler parts. This engineering design method was used to break down the logistical system into four primary functions: retrieve, process, store, and output. From this, a cross reference table and hierarchical flow chart were made to depict the functions themselves and their sub-functions. The components of our system were then discussed in more detail and justified.

Discussion

System				
Function	Retrieve	Process	Store	Output
Collect data from compressor failure reports	+			
Collect a list of mandatory replacements	+			
Collect list of obsolete parts	+			
Determine relevent vsnegligible parts to include in this analysis		+		
Determine how failed and obsolete parts are replaced	+	+		
Determine what conforming parts need to be replaced	+	+		
Determine what additional parts need replacing due to obsolete part changes		+		
Determine the age of the compressor and each individual part	+			
Record the experience and expertise of the engineers into the solution	+	+	+	+
Create a database to catelog the chronology of compressor repair		+	+	+
Adapt for future products	+	+	+	+
Memoize data		+	+	
Display Bill of Materials			+	+
Display list of recommended replacements based on historical data		+		+
Output all of the data to an executive summary		+	+	+
Organize and streamline the process such that it is replicable	+	+	+	+
Track part and serial numbers throughout each manufacturing stage	+	+	+	+

Figure 1: Functional Decomposition Cross Reference Table

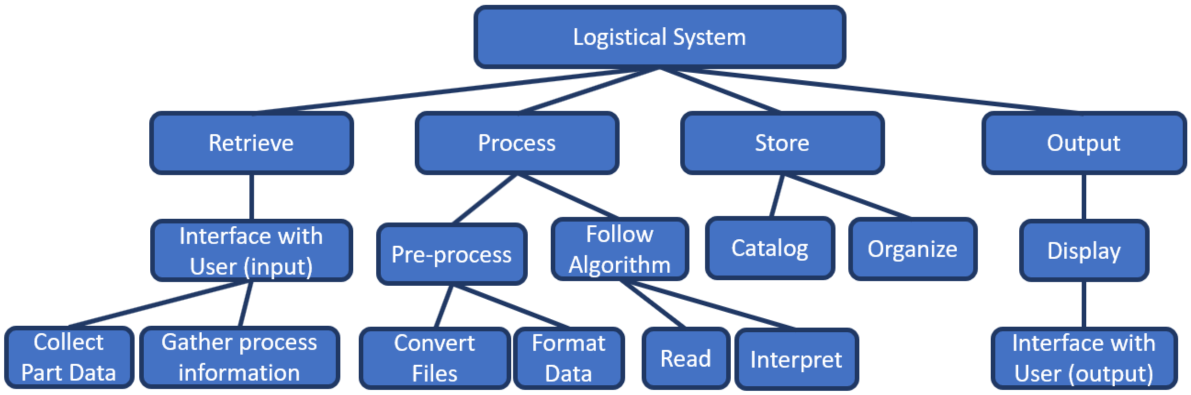

Figure 2: Functional Decomposition Flow Cart

From the information gathered in the customer needs, we concluded that the main four functions of the system are to retrieve, process, store and output the data coming in from failure analysis of the compressor. These four main functions will allow Danfoss to catalogue these parts. Further breakdown and analysis of tasks lead to the cross-reference table shown in Figure 1.

The first main function is retrieving the data. This function is responsible for all the inputs of our system and will be at the highest priority in the ranks of our system. The list of mandatory inputs for the system include mandatory replacement parts for the compressor, a list of obsolete parts, and the failure report data. All this information is necessary before entering the processing function of our system.

The processing portion of our system will be responsible for decision making. Ultimately, our system needs to be able to reflect the decisions made by the Danfoss supervisors. The supervisors want our system to be able to determine parts that have passed inspection, failed, and/or are obsolete. Since the system is iterative, there needs to be a storing function within the system.

Storage of data is required in the retrieval stage all the way down to the output of our system. The storing function is responsible for storing data to a database with all the information needed to catalogue these compressors. This includes, but is not limited to, serial numbers, age of the compressors, and other relevant information desired by the user.

The last main function that our system must execute is outputting the processed data. This data will be brought to technicians so that they have the information needed to repair the compressors. Therefore, the data should be user friendly and easy to use.

With all functions defined we analyzed how each function relates to one another. While the main functions start with retrieval and end with output, they step into one another with the defined minor functions. A good example is how in Figure 1 the minor function where the system is to record the expertise of the engineer is active throughout each major function. Moreover, the system provided is to be updated constantly. This results in the adaptation function, which also runs within each major function. While most minor functions can be traced down to more than one major function. Some of the minor functions can only be achieved while one major function is active. For example, all the collection of data from existing parts are to be retrieved in the first step as per the user's input request.

Justification

When coming up with the functions for the project we revisited our key goals and the customer needs to define what the system has to do. The main functions were defined as retrieve, process, store, and output. The major functions were then split into minor functions. Figure 2 illustrates how these functions were broken down into sections starting with the system as whole. The system was labeled "Logistical System", as its goal is to automate and streamline the aftermarket compressor repair process, taking the input from the users and giving them the desired output.

These four functions were chosen because they represent the needs of the customer in the most precise way. The system will retrieve the relevant failure data, process this data to determine what parts should be fixed, store this data in a local database, and output this data to the user. The system is designed to output data in a way that is easy to recognize for the user. The system also allows for a very seamless user interface whereby the user can input the relevant information in order to get the desired output. Since the system is designed to reflect the experience and expertise of the engineers, it delivers on storing and outputting this data in an organized way. Danfoss's primary problem with their current system was data management and lack of organization. This system would help to organize and catalog all relevant information related to the aftermarket repair process. This information can be retrieved and called upon at a later date in an automated way to significantly reduce the time required by the production planners. In order to satisfy the need of the system being able to adapt to future updates, the system is made with a user manual explaining all of the details of the operation of the system. The system is also made in a way that allows for changes to the input files and covers a wide range of potential part failures. This ensures that the system is functional and easily updatable as technology progresses and more failure data is added to the database. The functions take care of solving the intricate problem of keeping track of multiple part failures. It also determines what to replace each of those parts with while maintaining simplicity in the way it interfaces with the user. It is general enough to include all of the potential part failures, but limited in scope enough to know which parts are relevant to the compressor repair program. It is able to track the changes throughout the chronology of the compressor repair process and catalogs this information systematically. By performing these tasks, the system satisfies all of the customer needs.

Conclusion

Functional decomposition serves as a method to break down a complex system into smaller simpler parts. For this project we were able to break down the functionality into four main functions. The function in question became retrieve, process, store, and output. Based on the project's description, goal, and customer needs the four functions were then subdivided into the simplest task the logistical system is to achieve. With the functions defined, Figure 1 and 2 were created to better illustrate and represent what the logistical system has to do. To be clear, none of the defined functions specify a solution or a component. The functions are all tasks for the logistical system. Discussion of data serves as an explanation of each main and minor function, having Figure 2 represent a simplified connection of all functions to the main system. Figure 1 goes a step further by showing how minor functions associate with more than one major function. Justification of data references as to how the functions were defined and overall serve the goal of the project.