Homework 3

Assistant for Recycling and Waste Management to Reduce Recyclables in Landfills

Noah Beach, Kevin Christensen, Emmet Cooke, Jacob Leno, Caleb Scott
CS361 Group 18 - Spring 2018

1. Contributions Summary

The work was divided among our group for HW3 as follows: Noah Beach - Failures Mode Identification/Architectural Decomposition/Verification of Architecture, Kevin Christensen - Quality Attribute Identification/Architecture Validation, Emmet Cooke - High Level Architecture Diagram/Verification of Architecture, Jacob Leno - Failure Mode Identification/Architectural Decomposition, Caleb Scott - High Level Architecture Diagram/Quality Attribute Identification/Architecture Validation, Formatting.

2. Architecture Diagrams and Discussion

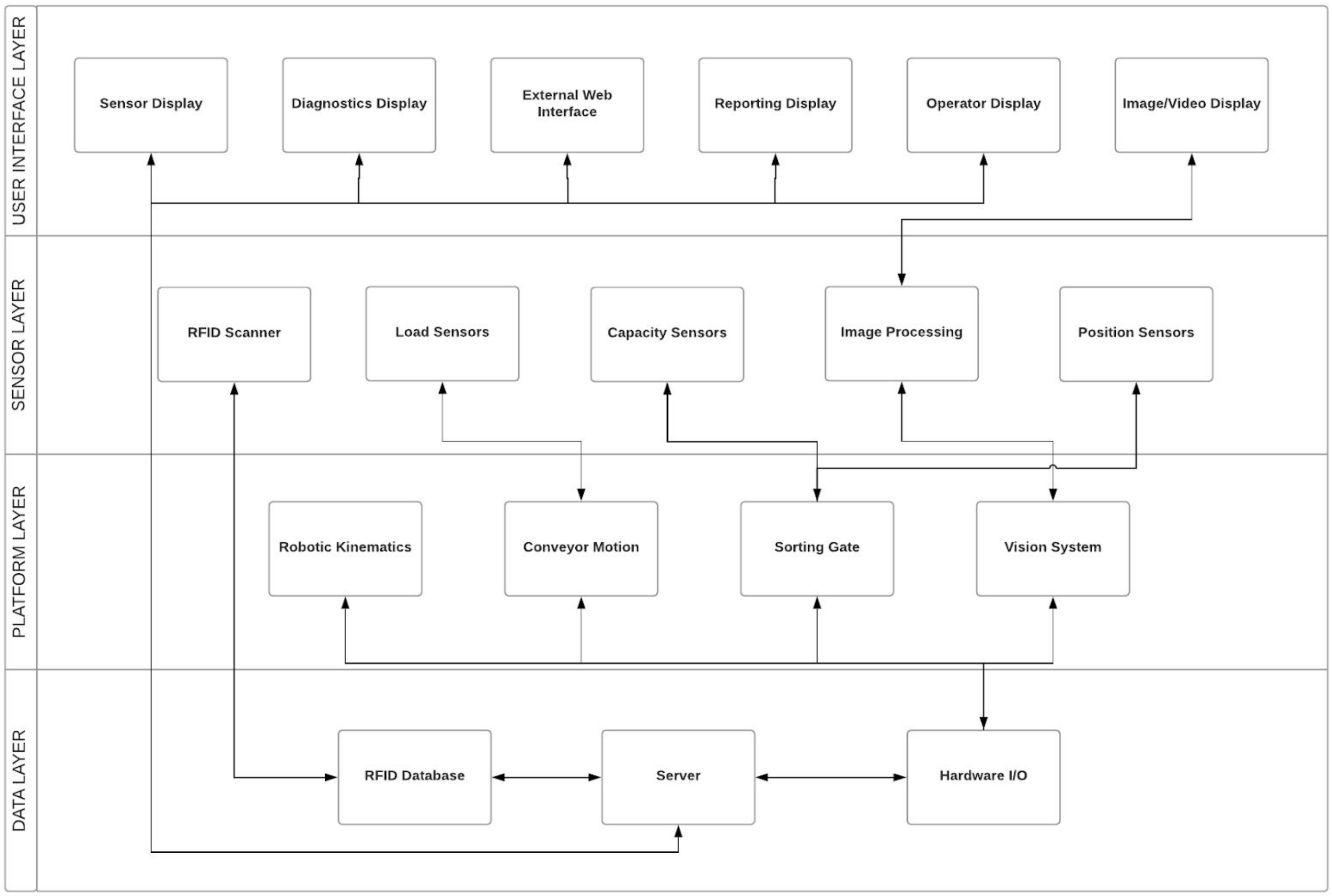
2.1 System Architecture

Selecting a system architecture is key a component to successful software design and execution. The architecture selected will determine how the many aspects and components of the software system are tied together and interact. Two potential architectures were selected as possible choices for the Recycling Assistant system, these are the Layered architecture and the Pipe and Filter architecture. These two architectures are discussed in more details in the following sections, as well as are represented via dataflow diagrams.

2.2 Layered Architecture

The Layered Architecture is composed of “layers” of services that abstract away the lower layers. This method can be used to model the Recycling Assistant system by diving the system up into four separate layers; the “Data Layer”, the “Platform Layer”, the “Sensor Layer”, and the “User Interface Layer”. The following diagram (figure 2.2a) is dataflow diagram of the Recycling Assistant system represented as a Layered Architecture.

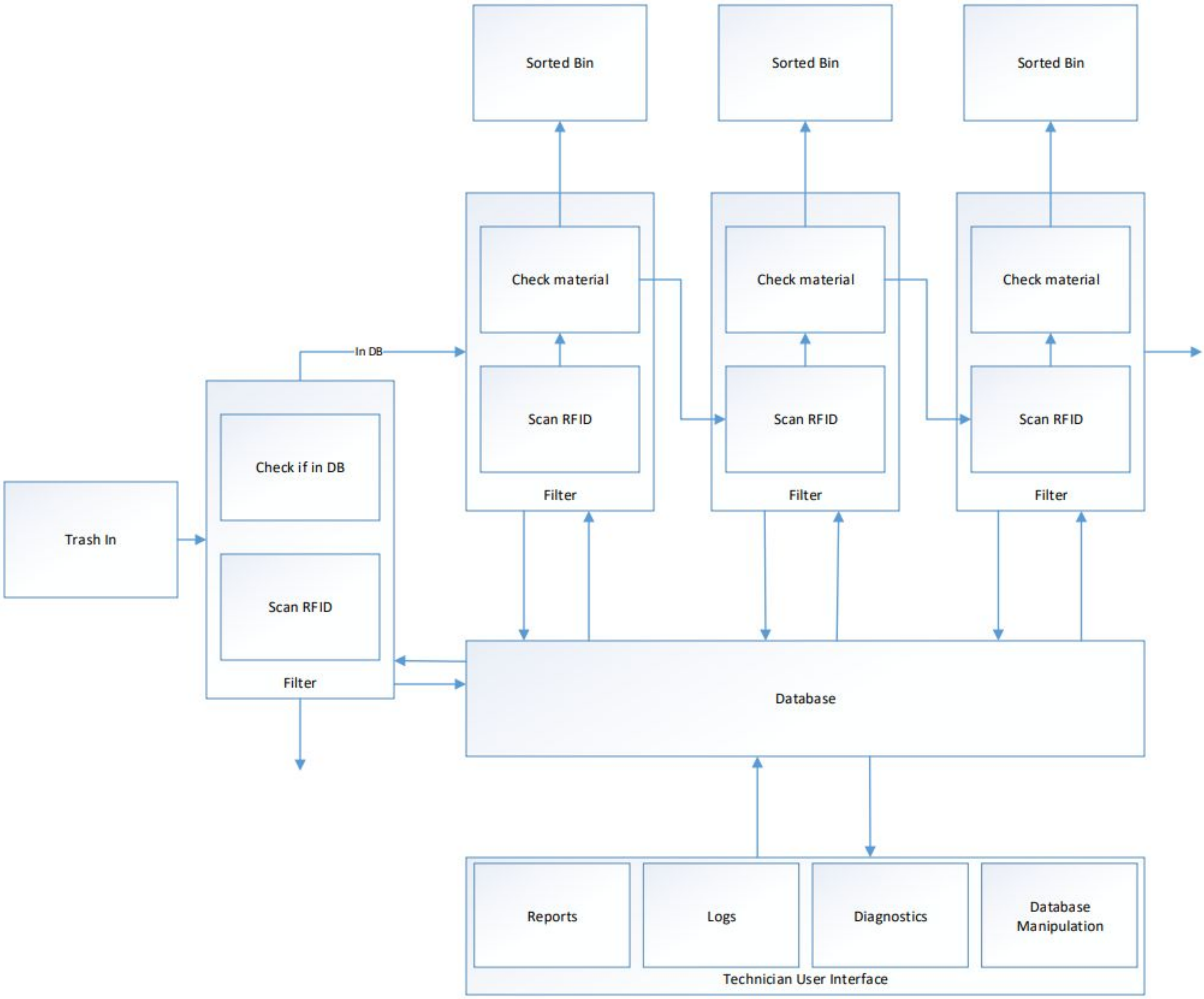
Figure 2.2a - Layered Architecture Dataflow Diagram:



2.3 Pipe and Filter Architecture

The Pipe and Filter Architecture is used to represent a system as a series of pipes that connect different components (filters) that alter or change data or objects that pass through them. The following diagram (figure 2.3a) is dataflow diagram of the Recycling Assistant system represented as a Pipe and Filter Architecture.

Figure 2.3a - Pipe and Filter Architecture Dataflow Diagram:



3. System Qualities Analysis

3.1 System Quality Attributes

3.1.1 Maintainability

- Can the database be updated with new information?
- Is there enough I/O for hardware expansion?

- Can controllers be updated without physically opening the devices?

3.1.2 Reliability

- Can the system run indefinitely?
    - Potential for 24/7 operation
- Can the controller properly differentiate between trash and recyclables?
- How does the system handle damaged RFID tags?
- Can false positives be avoided?

3.1.3 Efficiency

- Can the system react quickly to variable throughput of waste?
- Be able to turn on/off and control additional sorting capabilities to keep up with current demand
- Can the system reduce sorting time through recognition of common items?
- Can the various sensor polling rates be optimized to minimize data rate/volume?

3.1.4 Flexibility

- Can the system easily be scaled to include additional controllers?
- Establishing maximum system throughput.
- Can the communication bus handle the maximum number of controllers?
- Can the communication bus handle the maximum transmission length?
- Is it possible to add/change what is allowed to be recycled?

3.1.5 Testability

- Can the sorting results be easily tested?
- Are there systems to test for false positives/misidentifications?
- Can the system test for false or corrupted data?

3.1.6 Usability

- Can the technicians get the required diagnostic information to quickly resolve issues?
- Can reports of waste throughput be easily accessed?
    - Are city recycling employees able to generate accurate statistics of the amount/kind of recycling being processed by each center
- Can the system report recycling statistics to the general public?

3.2 Quality Attribute Assessment of the Layered Architecture Model

The layered architecture model presented is a broader, higher-level overview which allows for greater flexibility as the specific details of each process are not defined as much as they are in the pipe and filter model. The system should be able to be scaled to fit the requirements in terms of recyclables and controllers. The communication bus was addressed as hardware I/O and should be sized appropriately.

In terms of maintainability the database and hardware I/O update processes were not specifically addressed. Until the controlling hardware is determined the ability to be remotely maintained is also unclear at this stage in the design.

Reliability is better represented in terms of having all of the sensor data available in addition to having the displays for both the sensors and the diagnostic UI for the technician. Having this kind of data available will lead to a more reliable system as it will be able to be kept in optimal shape easier and issues can be detected and analysed before catastrophic failures. Additionally by having a vision system the system will be able to review and reject false positives as well as have a second look at waste with damaged tags and to double check all the waste coming through to reduce false positives.

Efficiency is an important requirement for this kind of system that deals with huge volumes of both physical waste that needs to be catalogued and sorted but also data in terms of multiple sensors polling, a vision system as well as the database and web application. All of this simultaneous transfer of information and waste will require a system to be efficient enough to handle both the maximum loads and data throughputs. The problem is the ability efficiently handle this kind of load is difficult to assess from this form of dataflow diagram.

Testability is the weakest attribute for the layered model due to the fact that it is a high level overview that doesn't dive into the specifics of each process. Without that level of detail there is nothing laid out to show where that kind of automated testing is planned.

The usability aspect is covered as shown in the user interface layer on the layered model. There are 3 distinct user groups for the system, the diagnostics and sensor screens for the technician, the reporting display for the recycling employees, and the external web interface for community members to see recycling statistics.

3.3 Quality Attribute Assessment of the Pipe and Filter model

The pipe and filter model defines the system architecture in a fairly linear fashion. As data (waste) enters the system boundary, it goes from filter to filter where the waste is either passed through the pipe to the next filtering controller or removed from the stream.

For maintainability, this architecture doesn't present many advantages, but it also doesn't offer disadvantages either. Much of the maintenance qualities are dependent on the hardware, with the exception of the database. The database in this model should be easily updated with new information through the UI that interfaces with it.

For reliability, this architecture should perform well. There's no reason to believe that each filter in the system would be incapable of continuous operation. Additionally, the nature of the pipe and filter model means that each filter should be able to properly differentiate between trash and recyclables and sort them properly. For handling damaged RFID tags, this architecture should be able to catch them and filter them out to hand sorting. This architecture should also be capable of preventing false positives.

This architecture suffers a bit with regard to efficiency. Due to its linear topology, it would not be able to quickly turn on additional sorting lines if a large throughput of waste were detected. It may be able to support the recognition and quick filtering of multiple common items. It should still function well to optimize sensor polling rates.

The pipe and filter architecture is decent for testability. Sorting results could be tested, though maybe not easily. Each filter could serve as a test for the previous filter if sorting data is shared between controllers. This architecture doesn't do well in testing for false positives as those items will be filtered out of the waste stream, either to the hand-sorting process or into a recyclable item bin. This architecture could test for false or corrupted data through data checksums or some other data integrity check.

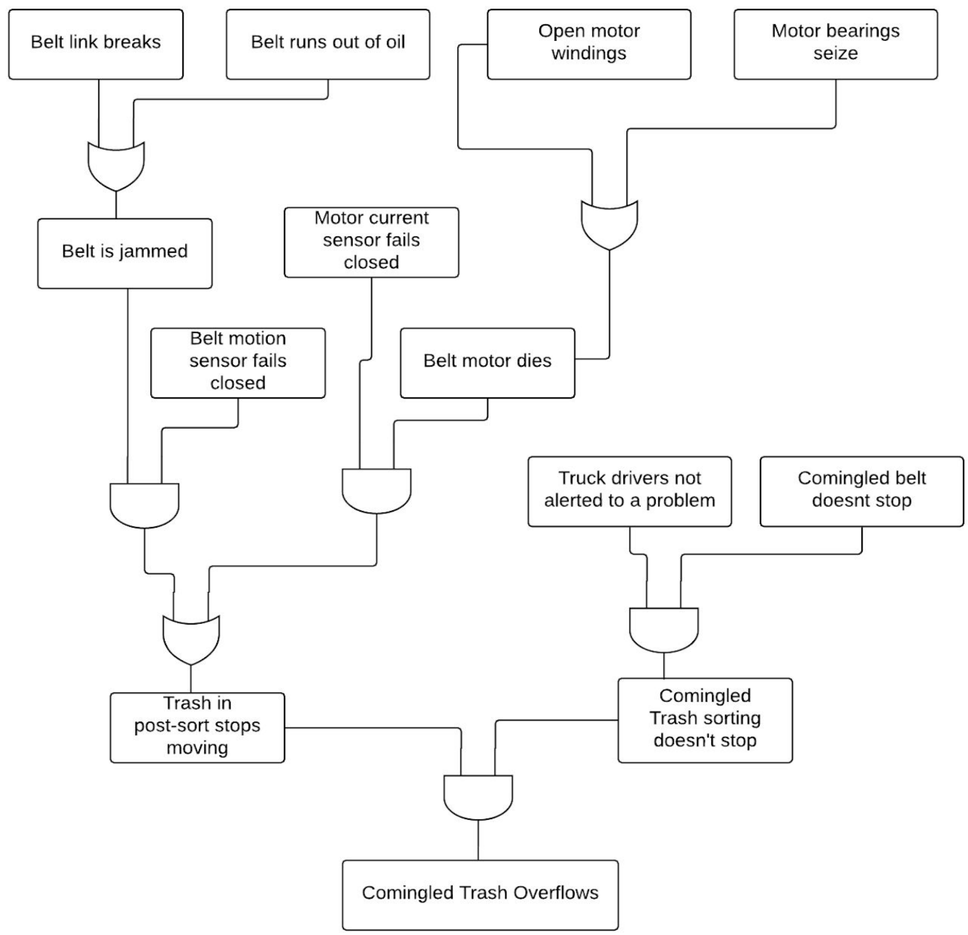
Finally, usability should be good with the pipe and filter architecture. Since all controller data goes through the database, retrieving it for diagnostics and reporting would be as simple as connecting to the database and requesting the information.

4. Failure Mode Analysis

4.1 Commingled Waste Overflow

One failure mode that was considered was the possibility of an overflow of commingled waste. This condition can be described in common vernacular as too much waste being loaded for which the system is able to keep up. Since waste collection facilities deal with very large quantities of recyclables this could be a serious issue. This fault condition explored further in the following fault tree (figure 4.1a).

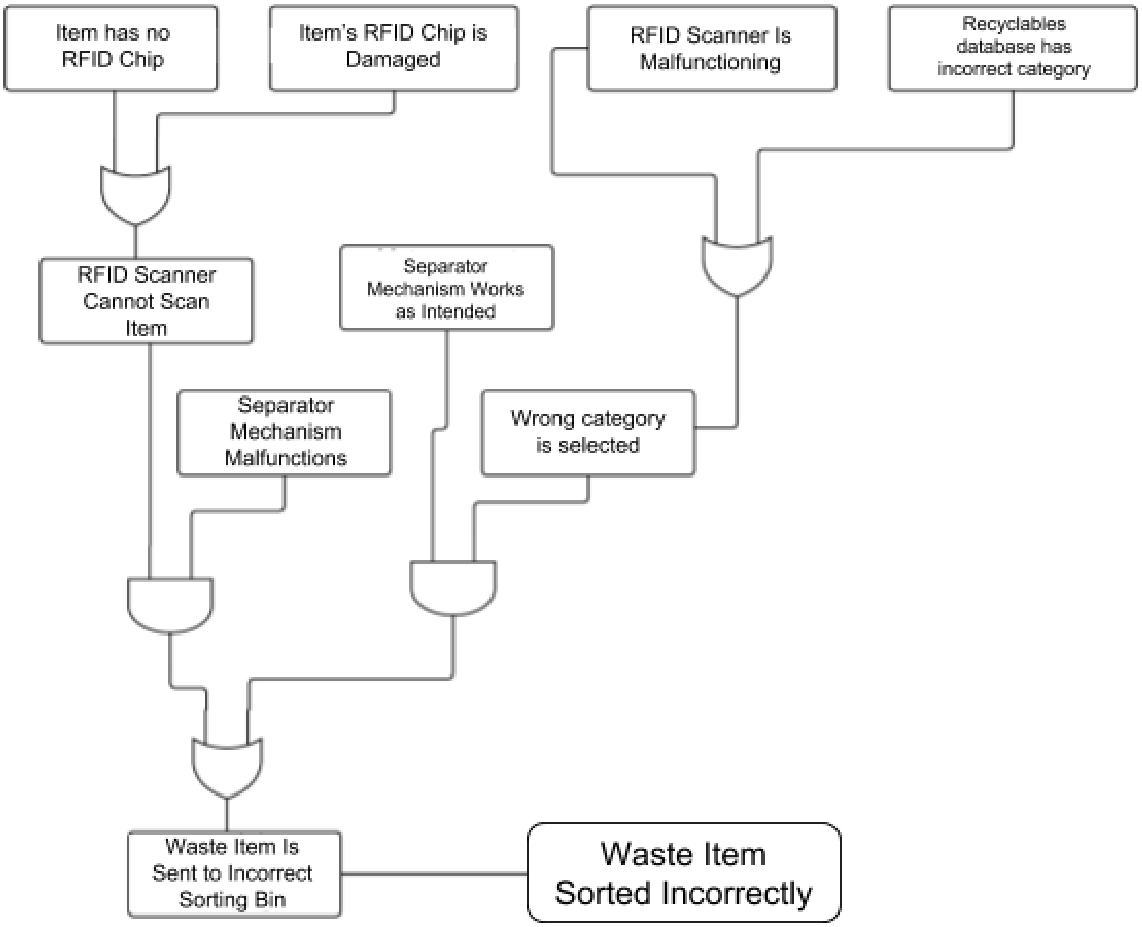
Figure 4.1a - Commingled Waste Fault Tree:



4.2 Commingled Waste Sorted Incorrectly

Another failure mode that was considered was the possibility of a missorting of commingled waste. This failure would result in an item being sorted incorrectly and ending up in the wrong final sorting location. This fault condition is explored further in the following fault tree (figure 4.2a).

Figure 4.2a - Commingled Waste Sorted Incorrectly Fault Tree:



4.3 Architecture Failure Mode Comparison

Based on the analysis of the two failure modes presented above, the more resilient of the two architectures compared (Layered vs Pipe and Filter) appears to be the Layered Architecture. Since the layered approach allows substantial abstraction of data and objects between the different "layers" of the architecture more resiliency can be built into the system (via sensors, additional software checks, etc) without adding the additional design overhead of incorporating those changes into all aspects of the system. As an example, the layered approach of the sensor layer alone provides significant advantage in design by allowing additional sensors to be added or sensors to be removed as the system changes and/or expands without requiring a redesign of the system as a whole.

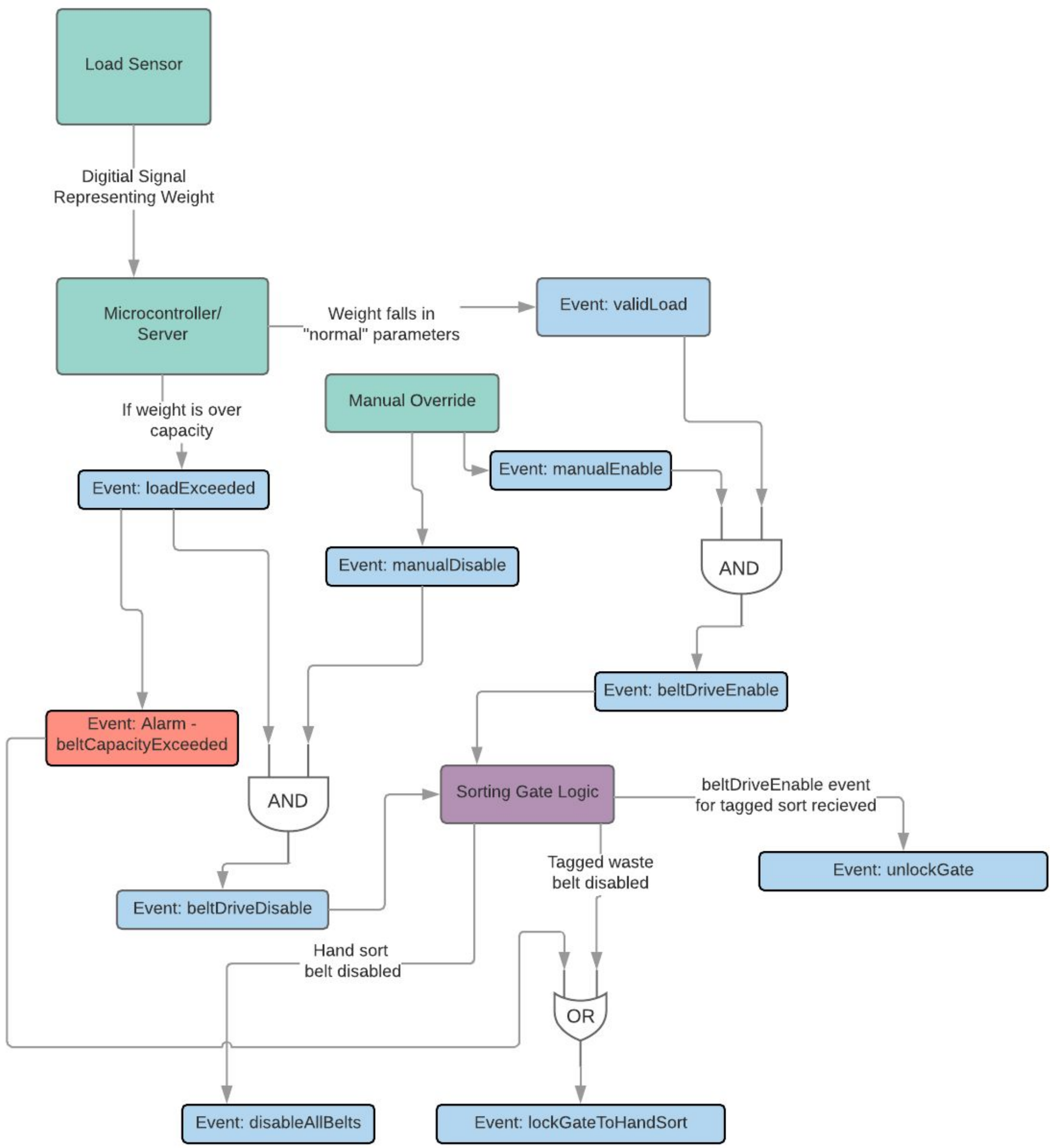
Considering the above, the Layered Architecture provides the most resilient design for the Recycling Assistant system.

5. Architecture Decomposition

5.1 Load Sensor Decomposition

One of the key components of the system design are the load sensors. The load sensors are positioned to be used to interpret if the conveyor belt has waste on it, as well as data concerning that load, such as its current relation to the total load capacity of the system. The load sensors are further decomposed in the following dataflow diagram (figure 5.1a).

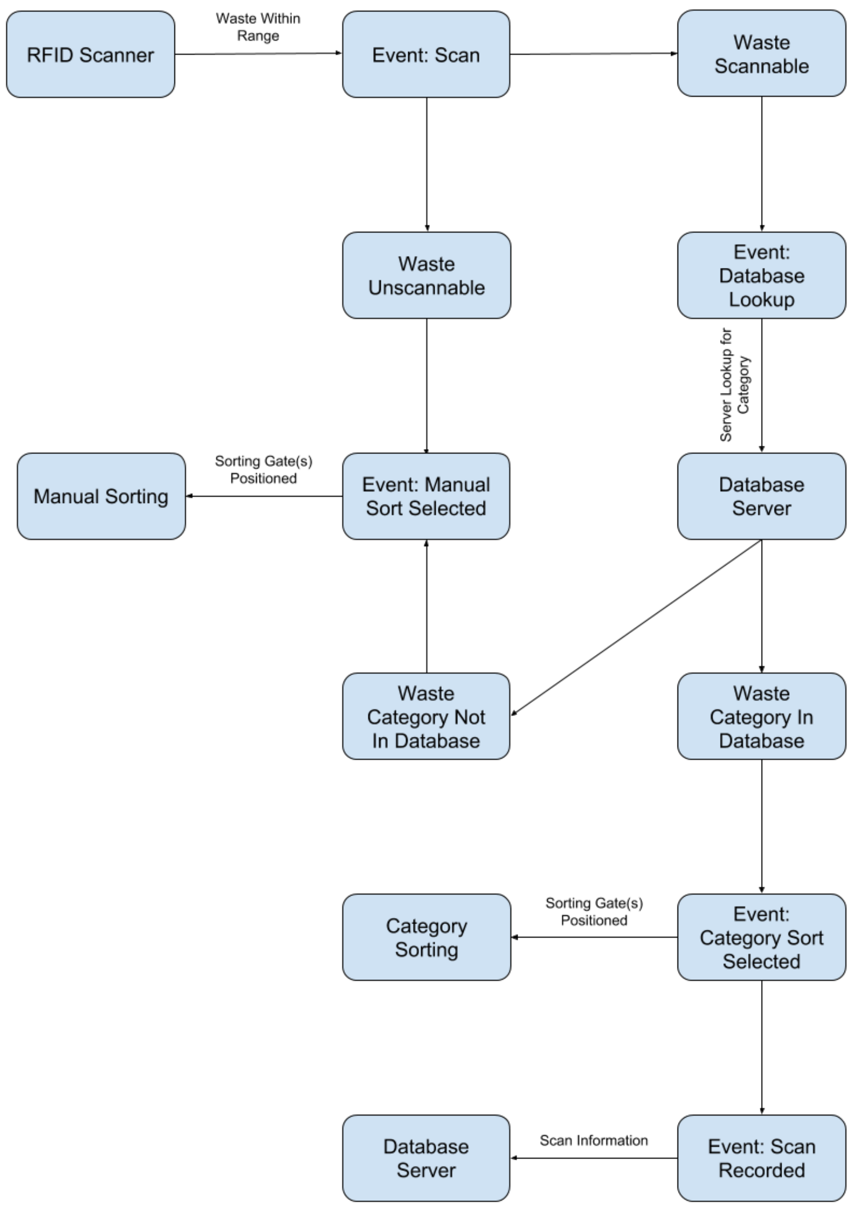
Figure 5.1a - Load Sensor Dataflow Diagram:



5.2 RFID Scanner Decomposition

Another key component of the system design is the RFID Scanner(s). The RFID Scanner(s) are positioned to be used to scan waste as it is moved through the system and determine if a scannable RFID tag is present in each piece of waste. If a tag is found, the RFID Scanner then interprets the data in that tag to a queryable RFID tag ID, which is then used by the system to analyze the waste for sorting. The RFID Scanner(s) are further decomposed in the following dataflow diagram (figure 5.2a).

Figure 5.1a - Load Sensor Dataflow Diagram:



6. Architecture Selection and Validation

6.1 Architecture Selection

After considering the analyses performed in section 3 (System Qualities Analysis) as well as section 4 (Failure Mode Analysis), the Layered architecture will be chosen as the selected architecture for the Recycling Assistant system.

6.2 Architecture Validation

The layered architecture model chosen specifically for the primary landfill location use case as outlined in Homework #2. The reason this specific model was chosen was done primarily due to the fact that the group is dealing with a large and complicated hardware and infrastructure development project that all has to be managed with software/firmware and databases.

Additionally there will be a web application component to the project which means the solution is very multifaceted and dealing with many different kinds of technologies both hardware and software.

Primarily this architecture validates the landfill use case because it lays out the overall strategy of setting up the landfill plant. Since it is a high level overview it is the perfect way to describe and begin to layout such a complicated system. With so many moving pieces relying on data or input from others it can seem like a daunting task to begin standing it up piece by piece.
As the recycling waste is being brought into the recycling assistant facility the RFID scanner on the sensor layer is scanning and categorizing the recycling by communicating with the RFID database and comparing the scanned tag with values stored in the database, the data is also passed to the controlling server so it will be able to sort it all down stream. From there the recycling waste will pass through the vision system for an additional scan to improve accuracy as well as to try and rule out false-positives or handle waste with damaged/missing RFID tags. Additionally once the recycling has been flagged the conveyor will bring it up to the sorting mechanism which will activate and move the sorting mechanism to move the pieces of recycling into their final destinations.

All of this is being monitored from the operators control panel. This will have all of the controls as well as diagnostics and reporting capabilities. Finally the external web interface is pulling data from the server which will allow members of the community to view the statistics of the recycling plant.

This all validates the model as all of the critical functionality required by the use case is included with the addition of a vision system for additional redundancy.

For the Individual Pre-Sorting use case, this architecture would work well. For example, in the use case, a user brings their item near the recycling bin opening. The scanner scans and determines if the item is recyclable. The vision system would be needed to determine if an item is present that doesn't have an RFID tag. As such, both the vision system and RFID scanner would operate in unison independent of one another. Data for both is sent to the embedded controller to process the data and command the various hardware peripherals. With the architecture, all of this is included, supporting the use case.

Finally, this architecture also supports the Transportation Sorting use case. In the use case, the system uses a robotic arm to pull recyclable items identified with the RFID scanner out of the waste pile. In the architecture, a conveyor and gate system is used in addition to any robotic sorting. The use case describes a system that takes items out of the waste stream and sorts all in one step. For this use case, the conveyor motion aspect of the platform layer would be left unimplemented, but everything else would still be used in the same way that the architecture describes.

Implications

After performing a validation and verification of the selected layered architecture the analysis shows that no major modifications should need to be made. Small modifications will be required, one such is the handling of the conveyor motion in the Transportation Sorting use case, but given the ability to perform abstraction with the Layered Architecture, this should be easily remedied. The analysis also shows that further details should be drawn up to diagram and analyze the vision systems.

The selected architecture aligns well with the quality attributes for the system. An example is when dealing with maintainability and usability attributes which are met due to the layered approach and the abstraction it brings. As an example, a user can interact with the user interface portion of the system, User Interface Layer, without requiring deep knowledge of any other part of the system. The interface can be changed to increase usability, or functionality without redesign.

The layered architecture also meets the requirements for reliability and efficiency, allowing sensors and parts to be replaced or potentially swapped with different models or parts than originally designed, without the upper and lower layers of the architecture being affected. This feature is also directly applicable to the flexibility quality attribute, and its requirements of scaling and replacement.

Finally the layered architecture will make testability, a quality attribute, possible by allowing each layer (and the components contained within) to be tested independently of other aspects of the system. This will increase the amount of tests that can be performed, resulting in a less error prone and therefore more reliable system, thus limiting failure mode occurrence.