

Ricardo Garcia Rosas 643952

The main issue I encountered while implementing the different strategies and trying to implement more complex strategies, was the lack of information provided by the interfaces. Using the summaries as the main source of information is a good implementation that avoids memory leaks; however, the information provided in the summaries is very basic and complicates the implementation of more complex algorithms that take the advantage of more system information to organize the mail for better performance.

Another shortcoming of this implementation is the creation/handling of storage boxes. Having boxes that get “destroyed” every time that a delivery bot takes it, complicates the implementation of methods that use indexing or sorting by floor numbers. As well as having a volatile number of boxes. A better implementation could allow these boxes to be organized in particular way to facilitate sorting.

Not having any documentation on the flow of the program makes it more difficult to understand the process/behavior that is taking place. A lot of the time gets spend trying to understand what the code does and how it behaves, rather than designing and implementing new strategies.

The abstraction of the software into functional units such as storage boxes, delivery bots, storage room, etc. helps understand the overall behavior of the system because it is relatable to actual physical system units. And the usage of interfaces to define the interaction between these units/classes has the advantage of facilitating the implementation of different algorithms that are suitable for different situations, and that can be selected depending on it. Unfortunately the selection of these is performed in the Simulation driver, rather than by the system itself. Making the system not adaptive to different scenarios.