CS554 Human Computer Interaction

Assignment 2

If we describe the sequence of steps, to open an attachment file in OtherApplication from EmailReader:

- The user selects the attachment to be opened.
- Application selection
  - either default application automatically selected for the file format
  - or choice of applications for the file
  - or file format not recognized.
- The attachment opens in OtherApplication.

The Interaction framework described by Abowd has 4 steps to analyze a task. The problems that can arise in each step are detailed below.

- **Articulation** : When the user translates the task to a set of actions to the system.

  - **The visibility of attachments**: The User Interface of the app should be familiar, and easy to navigate. The placement of the section where attachments are displayed is important. If they are not immediately visible to the user then that's a problem.
  - **Mutiple attachments**: When dealing with multiple attachments(potentially large number of them), the interface can encounter problem. Finding the particular attachment from the list, should be considered while designing.
  - **Application to open the attachment**: User may have to specify the application to open the attachment in. For this, the mail application needs to show some kind of context menu or popup menu to facilitate selection. This needs to be carefully handled as displaying every application for a file, may not be helpful. Instead only applications on the OS that are known to handle that file, may be displayed.

- **Performance** : When the system performs the task in the core language.

  - **Fetching the attachment**: This involves use of network resources. A problem can be when the attachment is very big in size, and the network gets interrupted.
  - **Saving the attachment**: This involves use of storage resouces, to temporarily store the attachment file to display. The space needs to be present to store these files.
  - **Connecting with other applications**: This typically involves using OS resources(API) to fetch a list of installed applications that can open a particular file. The applications need to internally communicate for the file to be displayed.
  - **Corrupt Attachment**: The file sent can be potentially corrupted with missing data or have an unrecognized file format. This needs to be handled properly, as it can risk crashing the opening application.
  - **Malicious Attachment**: The file can also be a malicious executable, which on many OS can be a big security vulnerability when directly executed. Such file formats need to be handled correctly.

- **Presentation** : When the system translates its internal state to the user.

  - **Displaying the state of attachment**: This occurs when the user selects the attachment to be opened. The system should appropriately display the state of the attachment. These states can be:
    - **Downloading** - where the mail app is downloading the attachment from the mail server.
    - **Downloaded/Saved** - where the attachment is saved on the user's system. A further usecase can to delete the saved attachment to free up resources.
    - **Opening** - where the mail app tries to opens the attachment.

- **Displaying Error messages**: Handling errors encountered in the performance phase, and next action for the user to take.
- **Displaying choices**: Displaying the choices for opening file, applications that are currently present in the system, and in context to the type of file.

- **Observation** : When the user interprets the system actions to see if the task was accomplished.

  - - **Feedback**: A confirmation can be given to user with a sound alert or popup notification, about the success or failure in opening the attachment.
  - **Application transition**: When the transition to OtherApplication is not apparent on the screen; when started minimized, or in the background. User can get confused in this case, without getting any feedback.
  - **Application error**: The application which opens might be corrupt, or crash when being run by the EmailReader. The reason may be related to the file or it can independent.