In our chaotic world of ever expanding monitor sizes and increased demand for multi-tasking, we are often confronted with a plethora of windows and no logical way to organize them. When working on a paper, a user may be typing in a word processor, looking at several web pages, some PDF documents, and a video of related information. He may also have a music player, an email client and a calendar application open. When the user decides to switch from working on his paper to checking his email, he must hunt through layers of windows in order to find the correct application. Solutions to this problem exist, such as workspaces in UNIX and Spaces on Mac OS X, however they require manual organization of windows by the user. I believe this procedure is unnecessary and hinders productivity.

In the above example, all of the windows are related through contained textual and/or media content. I suggest that data mining techniques could be used to cluster windows into groups of related information or ‘tasks’. The user could then find the window they were looking for by switching to the appropriate task. In order for this to be possible, it is first necessary to parse
data from a window and process this information using a clustering algorithm. These sorted ‘tasks’ must then be presented to the user in an intuitive and useful way.

**Schedule**

I will divide my time into the following milestones (detailed later in document):

- Preparation: meeting with professors, developing idea, getting project approved (start of semester – Feb 8th)
- Research (Feb 9th – Feb 21st)
- Design (Feb 22nd – Mar 6th)
- Implementation (Mar 7th – Apr 17th)
- Polish and prepare for presentation (Apr 18th – May 5th)

I will tackle this project by working on two branches in parallel. The first branch is the clustering of windows into tasks and the second is the presentation of these tasks.

**Preliminary Research**

Easing task management on a computer system is a very modern field of research. Microsoft’s Human Computer Interaction group has done considerable work in this area creating a prototype dubbed SWISH (1). SWISH examines the ‘relatedness’ of tasks through analysis of the semantic similarity of window titles, and the temporal closeness in their access patterns. With 4 hours of user data, the team was able to achieve classification accuracies of about 70%.

There has also been considerable work into measuring the ‘relatedness’ of concepts and text. Measuring the semantic similarity of two concepts is possible through methods described by Ted Pedersen (2). His work is implemented in a Perl library Wordnet::Similarity and measures relatedness through information found in the Perl lexical database WordNet.

When it comes to the presentation of windows in related tasks, several solutions currently exist. Linux distributions such as Ubuntu use Compiz Fusion as a window manager. Compiz Fusion provides a number of desktop spaces that contain windows, which can be switched between using the cube-like interface shown in Figure 1. Mac OSX implements a very similar, albeit more elegant version of this concept called Spaces (shown in Figure 2).
While there has been research into clustering windows based on title and temporal closeness in access patterns, there has been little or no work done to measure the ‘relatedness’ of windows based on their content. In addition, once relatedness has been established, automated organization of tasks into separate containers for usability testing has not been attempted.

**Milestones – Clustering**

1. *Research*: Meet with Martin Schultz and Dana Angluin to discuss methods for clustering windows based on the following:
   - Time of instantiation
   - Window title
   - Application name
   - Application content (e.g. text of word document or HTML of web page)

2. *Research*: Extend knowledge of Gnome desktop environment and the Compiz Fusion window manager. These tools will be used to extract the data above.

3. *Design*: I will formalize an efficient method to extract the required data from each window in a form that can be processed by my clustering algorithm. The semantics of the clustering algorithm will be formalized after further research and discussion with aforementioned professors. The results from clustering must be in a form that can be passed to my presentation engine.

4. *Implementation*: Window data will be extracted using Gnome and Compiz Fusion. My clustering algorithm will be implemented using Python and R. The clustering process will be called manually by the user, when required.

**Milestones – Presentation**

1. *Research*: Extend knowledge of the Compiz Fusion window manager. Learn how to manipulate windows (moving them from space to space).
2. **Design**: Given the grouping of the windows computed by the clustering process, the presentation engine will reorganize windows into desktop spaces. Each space will represent one cluster or ‘task’.

3. **Implementation**: The presentation of groups will be implemented through manipulation of windows using Compiz Fusion on the Gnome desktop environment.

**Deliverables**
- Code used to parse the informational data of a window
- Python/R code for the clustering algorithm
- Presentation code for organization of windows
- Oral Presentation

**Works Cited**