Hypermedia API to access desktop capabilities

Alain Sanguinetti

January 28th, 2016

Abstract

The main goal of this project was to provide a simple interface, ressembling a book to allow access to files and also allow some interaction with those files. My personal goal in this project was to use HTML and HTTP requests to get more knowledge of this combination.

The assumptions made were that the user has the ability to click, scroll and enter text, provided by the client.

The system developed provides some browsing capabilities of common file types and some interactive capabilities. Developing the system was very instructive.

Class: M.S. program in Embedded Robotics
Instructor: W. Paluszunski
Department: Katedra Cybernetyki i Robotyki
University: Wroclaw University of Technology

1 Introduction

This project started with a simple observation. There is a discrepancy between the level of complexity of interfaces of personal devices and the level of understanding of the people using them. For example, many old people have troubles using smartphones or computers because they don't understand the interface. This is the problem that the project tried to answer. This project was to develop a simple interface for simple processing tasks: reading and answering emails, viewing pictures.

The most naturally understood interface to display information is a book. Books are understood by most of people, most of us can find information in a book using a table of contents, page numbers, titles and all the other signs. The computer version of a book is a simple webpage using HTML. It can present information in the same formatting as a book: references, lists, heading, paragraph, images. Because of this parallel between books and websites, it was chosen to implement this interface using HTML and HTTP requests.

The combination of HTML and HTTP requests is a form of hypermedium. This is what I learned while watching a video about "hypermedia API" [3]. This video really raised my interest but I had not a sense of how this really is implemented on projects.

The objective of the project was two-fold : to develop a simple interface and to play with the hypermedia concept.

In the following pages, we will look into what was developed and evaluate how much it met the goals of the project.

2 The proposed implementation of the resulting interface

2.1 Tools

The tools used to realize this project are Python 3 and a hierarical file system.

Python 3 makes it very easy to set up a simple server through the http package which is included in the base distribution of Python. The script defines methods to handle the HTTP requests GET, POST, PUT and DELETE. The script is able to display directories, txt files, jpg files and json files.

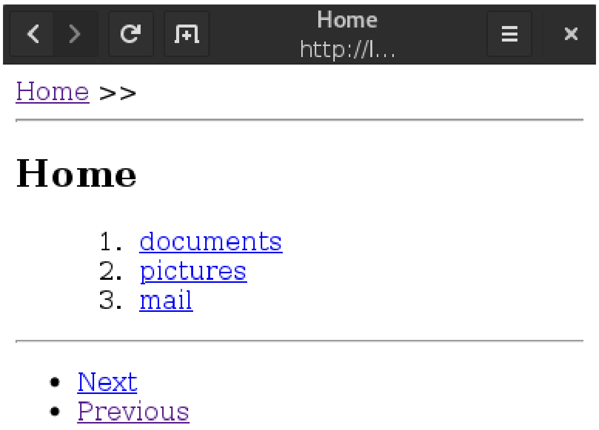
The files are stored in a hierarchical file system. This facilitates the analogy to the book because it is as if each folder is a section of the book, then it is easy to represent a table of contents as a list of folders.

2.2 Interface

The interface developed consist of a header with breadcrumbs, a menu, a body and a footer.

The content of those for sections is created depending on the file type.

The figure 1 shows a presentation of the root directory, the figure 2 shows the presentation of an email, the figure 3 shows the presentation of jpg file and the figure 4 shows the presentation of a text file.



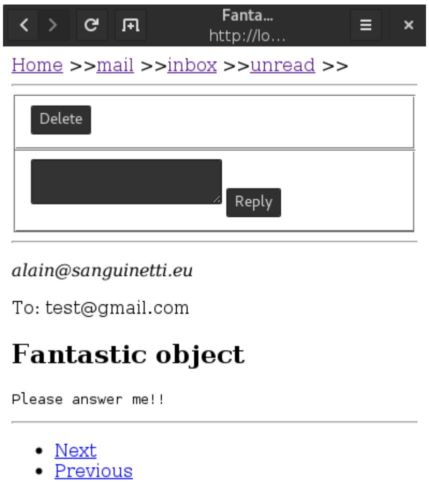
Home >>

Home

1. documents
2. pictures
3. mail

- Next
- Previous

Figure 1: The home page of the interface



Home >>mail >>inbox >>unread >>

alain@sanguinetti.eu

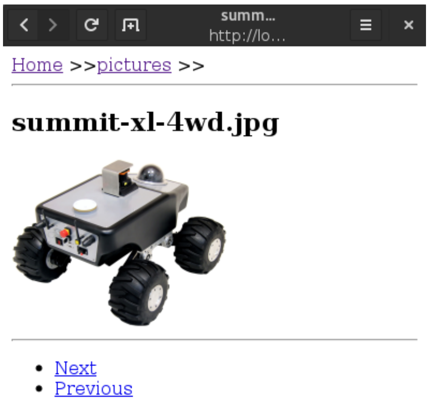
To: test@gmail.com

Fantastic object

```
Please answer me!!
```

- Next
- Previous

Figure 2: Reading an e-mail

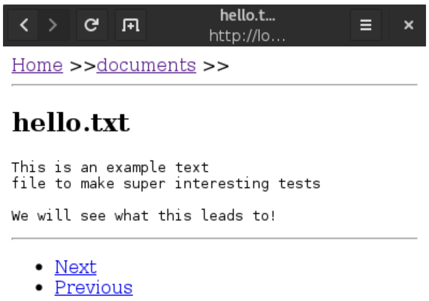


Home >>pictures >>

summit-xl-4wd.jpg

- Next
- Previous

Figure 3: The display of a picture



Home >>documents >>

hello.txt

```
This is an example text
file to make super interesting tests

We will see what this leads to!
```

- Next
- Previous

Figure 4: Display of a text file

2.3 License

The project uses Python which has a special license [2], this license is GPL-compatible. The developed project is not subject to any special license and may be used according to the french intellectual propriety law.

3 Evaluation of the results

3.1 Simple interface

The resulting interface is very simple, and has the appearance of a page of a book. Pages can be flipped using the "next" or "previous" links. In this way, the goal is achieved. However, the interface lacks functionnalities : for example it is impossible to write an email or to modify text.

In this regard, we can say that the project validates validates the usability of a simple interface but it has yet to be tested using a fully-featured interface and a panel of users.

3.2 Hypermedia interface

This project makes full use of HTML language and HTTP requests. Every component of the HTML page has a unique identifier given by the combination of tags that defines it. This makes possible the creation of a client that sends a request to the server and parses the HTML response to identify specific information. This capability was not demonstrated in the project. Also, one of the interesting features of the video[3] showing hypermedia API was the use of class names that refer to URLs of a documentation. This feature was not implemented here because there was no need of identifier, however this does not allow easy documentation.

In the regard of being a hypermedia interface to a desktop, the project succeeds in relying on HTML and HTTP requests and providing an easy interface for any HTML-HTTP enabled client.

3.3 Looking further

This project could be assembled in a linux executable that would be run as a service in the background. It would transform any computer into a remotely accessible filesystem, secured using HTTPS. The use of a simple HTML-API makes it easy to access it with a browser or to write dedicated apps to perform special tasks remotely.

Also, this project was interesting because it could be applied to a ROS system. Few days ago, ROS-control-Center[1] was released. It provides a web interface to access many information of a robot and also interact with it. However it does not offer the possibility to use it as an API because some elements are not unique or have reused class names. It would be interesting to create a control center that is just an HTML-HTTP gateway to a ROS system : it could be accessed from a browser as well as from another ROS system running anywhere else and talking using HTML-HTTP requests.

4 Summary

We wanted to implement a simple interface for desktop computing tasks and use HTML-HTTP as an API. Both goals were reached to a certain level, however the project can be regarded as a prototype that validates the use of those concepts together but not as a functional release. This project was nevertheless very interesting to fulfil and may be transformed into robotic tools in a near future.

References

[1] Lars Berscheid. *ROS Control Center*. 2016. URL: http://www.ros.org/news/2016/01/ros-web-control-center.html.

[2] Python Software Foundation. *PSF License for Python 3.5.1*. 2016. URL: https://docs.python.org/3/license.html#psf-license-agreement-for-python-release.

[3] Jon Moore. *Hypermedia APIs*. 2011. URL: https://vimeo.com/20781278.