

# Web Interface Based Operating System

Ms. Saraswathi Banoth<sup>1</sup>Ms. P Kavitha<sup>2</sup>

1. [M. Tech (CS)], Dept. of CSE, Ganapathy Engineering College
2. Asst. Professor, Dept. of CSE, Ganapathy Engineering College

## ABSTRACT:

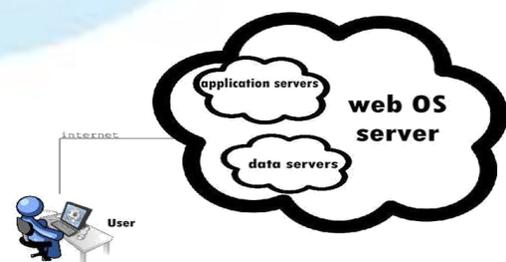
Web interface based Operating System is a web browser based Operating System which provides an environment that pretty much resembles a desktop environment through a web browser. It mimics the user interface of the computer operating system, but it doesn't interact directly with the computer's hardware, the users must still have an OS on their computer. The Web interface based Operating System employs a semantic file system which helps in better organization of data and a service oriented architecture where services are encapsulated to make them autonomous where they can be easily changed, deleted or added without influencing the whole system along with an inclusion of various software services.

**Keywords:** *Web interface based operating system, Semantic, Ontology, Service Oriented Operating System, SEWOS.*

## 1. INTRODUCTION

Web interface base OS can be referred as a virtual Desktop accessible through a web browser with multiple integrated inbuilt services that allow the users to manage their data from any location and is independent of the usual operating systems. The Web interface based Operating System employs a semantic file system which helps in better organization of data and a service oriented architecture where services are encapsulated to make them autonomous where they can easily be changed, deleted or added without in quenching the whole system. Web interface base operating system provides users with traditional operating system applications as services available for user to access transparently

without any prior knowledge about where service is available, the cost or constraints. In web interface base operating system, applications, data files, configurations, settings and access privileges reside remotely over network as services accessed by web browser which is used for input and display purposes.



**Fig.1:** Web interface based Operating System Architecture.

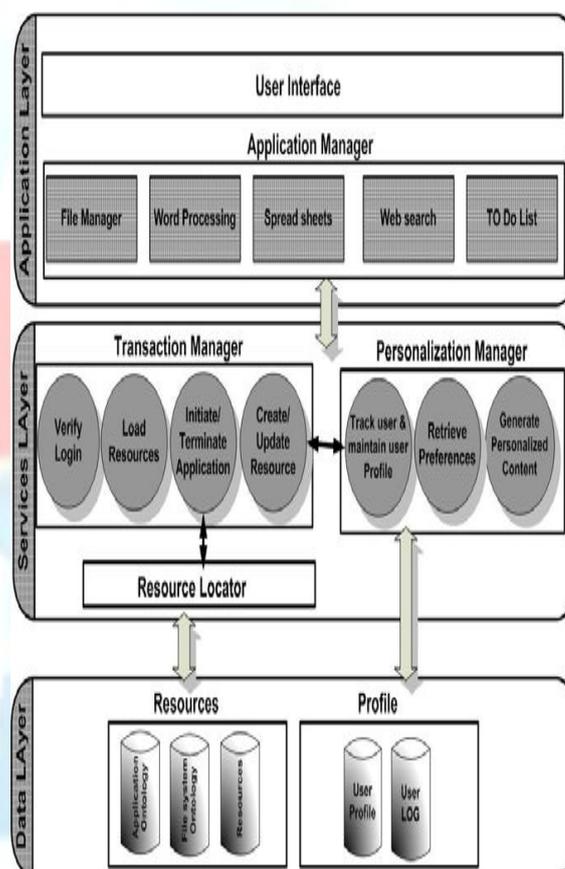
## 2. THE PROPOSED ARCHITECHTURE

Keeping one thing in mind, to develop a Web interface based Operating system that satisfies features and functionalities of a personalized desktop operating system. As discussed earlier web interface based operating system transfer applications to web server, where users can manage available resources through virtual desktop using web browser. In order to do so we had three main interests which we tried to satisfy.

- Switching from personalized desktop to web desktop is hard task, since users will not accept nothing less than their own personalized desktop that they are accustomed to. Therefore it was important to maintain that user will always have a personal experience that resembles his fully personalized pc environment.
- Semantic web technology plays a significant role in today's web as well as desktop systems. Thus since semantic technologies have not been used in any web interface based OS, it was just a matter of time before it becomes a research thrust.
- A service oriented architecture (SOA) is the next evolutionary step in building web-based applications as it provides a set of principles of monitoring concepts used during various stages of developing a system.

## 3. SEWOS ARCHITEHTURE

SEWOS is SOA based architecture that shows the semantic file system of the web operating system. This Web interface based operating system is based on this architecture. SEWOS consists of three layers, application layer, service layer and data layer as shown in the figure. Each layer works in correlation with one another to provide a good interface and a stable environment.



**Fig.2:** SEWOS Architecture

## 4. SYSTEM IMPLEMENTATION

SEWOS was developed based on SOA techniques. First, user must go through a registration process or an ordinary login for

registered users. Afterwards, user will be able to view his personalized desktop, access and manage his own resources and applications. SEWOS makes use of memorization as a personalization function, displaying a welcome message and a fills user's personalized start. Menu with his recent file list, his events, his favorite resources and applications. Besides his start menu, user can start any application directly using application icon on his desktop. Moreover, user can start and deal with multiple applications at the same time. Options to manage workspace preferences are also available and accessible through personalized desktop. The implementation of SEWOS home page and personalized desktop is shown in Fig 2.

#### 4.1 APPLICATION LAYER

Application layer contains user interface and application directory which includes set of applications: file manager, notepad etc.

User interface provides you with login/logout service. It also provides the user with desktop icons. One of the interactive applications includes the start taskbar embedded with clock and the current date. The home page personalized GUI that interacts with the user to open a services comes under user interface. User interface also includes the file browser, a GUI that provides list of files and folders along with there locations.

Application directory contents all the services

the web operating system provides the user with. They are notepad, mailer, compiler, music player and calculator. These services are applications that provide user with a platform to perform operations of their interest.

#### 4.2 SERVICE LAYER

The second layer is service layer which includes resource manager and personalization manager. This layer basically triggers backend operations.

Resource Manager controls the user requests and works along with data layer and application layer. It provides the application layer and data layer with an entity that they use while running a service. Login verification, Loading resource, application termination and creating a new resource are the services operated at the backend (behind interface). For example: a request to create a new folder is initiated, it would be the work of resource manager to create an entry in the directory.

Personalized Manager is responsible for generating a personalized desktop making use of user settings, preferences and profile. One of the examples being changing the user desktop background, saving sessions over logout and loading it back at the next login.

#### 4.3 DATA LINK LAYER

Data layer contains backend databases, which stores user profile, log file as well as user resources which are the user's personal files.

Data layer consists of resources; which are nothing but files associated either with services (like installation files) or user data files. Service files include the files used by application layer to run services and user data files are all personal files. At the same time resources keep user profile information like current background etc.

- **User resources:** as each user directory will consist of user's annotated resources.
- **Ontology:** file system ontology as well as application ontology.
- **User profile:** contains user's preferences and personalized settings.
- **User log:** keeps track of user's usage data, sessions and data storage.
- Data layer includes as well a set of work copies. As aforementioned, SEWOS presents transactional capabilities that help maintain users document and provide restoration capabilities' in case of application or connection failure.

#### 4.4 ONTOLOGY

Define the concepts and relationships used to describe and represent an area of knowledge

- RDFS and OWL can be considered as a simple ontology language
- Common knowledge and interests sharing within their community

- Tasks that ontology can be used to support a portal
  - Accessing a portal
    - Conceptual search and navigation
    - Inference capabilities
  - Providing information
    - Methods and tools accounting for the diversity of information

**User Registration:** The Registration Page allows a new user to register to the Semantic Web OS, it is written in PHP with Jscript. It is integrated with the users table in MYSQL and makes an entry into it given the user registers with valid information.

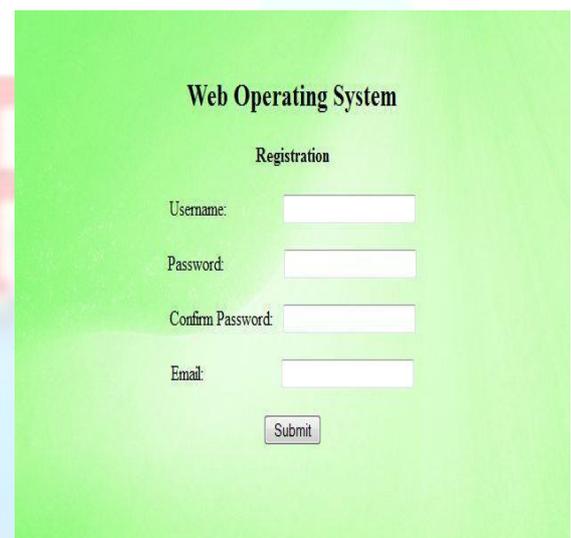


Fig.3 User Registration

**Login Verification:** The login page checks if the given user information is valid with users database in MYSQL. If found so, it allows the user to login into his/her personal space. The session variables are also initialized during this process which helps in loading user specific preferences and data.



**Fig.4** Login Verification

**Symantec File Index:** For every user an index.xml file is maintained that updates the details of files upon saving/modification and also maintains the entire directory structure for that particular user .Example: The figure shows the description of a user file system.

**Mailer and Notepad Services:** The Notepad service allows user to create files and stores it into the user directory. It also allows the user to read and modify previously saved files. The Mailer service provides an interface to send mails from the user's registered email id (given during initial registration with web interface based operating system) to various other contacts. Both these services are written in PHP and Jscript.

**File Uploader:** The file uploader is used to upload user files from the system onto the user directory of the web interface based operating system. The files are stored initially in a temporary file and then moved to the user specified location before the service

terminates. This service is written in PHP.

**File Explorer:** File Explorer gives the user an access to all his/her saved and uploaded files. It is used to traverse the entire user file directory and open the files with the supporting services. It also has added new folder and delete options to manage the files. Any changes made in the file structure are immediately reflected in the user index.xml file. It is written in PHP.

**Image Viewer and Music Player:** Image Viewer and Music Player are used to display images and play music whenever a JPEG/PNG and MP3 files are accessed respectively. The image viewer is written in Jscript and PHP, whereas the Music Player is written in flash action script.

**Service Directory:** Service Directory's serves as the service locator. It is an xml file that stores all properties related to all different services.

**Compiler:** This service allows users to write and compile/run their C/C++ files. It uses the server shell to compile and run the files. It also displays errors if the compilation fails. It is written using PHP.

```

localhost/services/compiler/index.php
# include <iostream>
using namespace std;

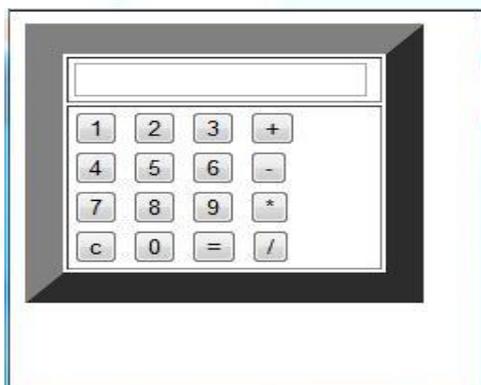
int main()
{
    int a=20;
    int b =5;
    cout<<a/b<<endl;
}
    
```

C++

Compile + Run

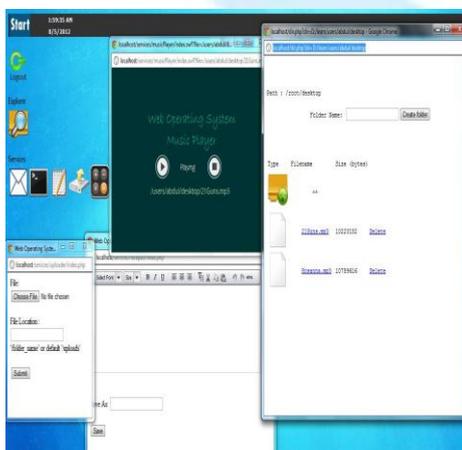
**Fig.5:** Compiler

**Calculator:** This service provides the user a simple calculator interface. It is written using HTML.



**Fig. 6:** Calculator

**Home Page and Services**



**Fig.7:** Application

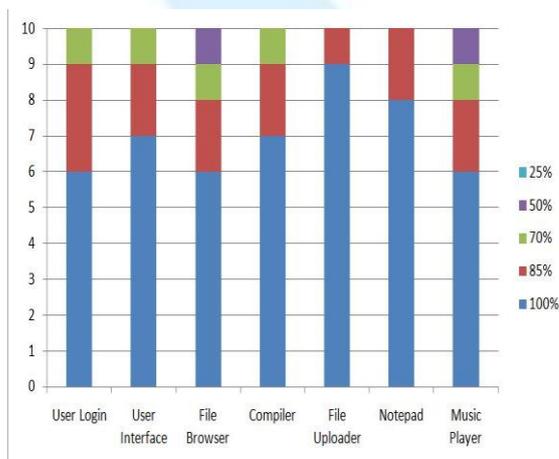
**4 Testing and Results**

Since this project is aimed at providing the users with a desktop like environment onto a web browser, the best approach for evaluating it was to conduct a small survey among different users. 10 students were chosen who were questioned on various features and services of the web interface based operating system. They rated each such feature from 1 to 5, each corresponding to a liking/satisfaction of 25%, 50%, 70%, 85% and 100% respectively. The survey revealed that for file browser, 60% of the users had 100% liking, 20% had 85% liking, 10% had 70% liking and 10% percent had 50% liking. On obtaining feedbacks we also learnt that the users expect to see copy, paste, move etc options with their files and folders. Similarly this survey was conducted on all features and a table containing the percentage liking towards different services is given below. Apart from these ratings the users also came up with various suggestions and features that they would like to see.

Features \ Liking	100%	85%	70%	50%	25%
User Login	60%	30%	10%	0	0
User Interface	70%	20%	10%	0	0
File Browser	60%	20%	10%	10%	0
Compiler	70%	20%	10%	0	0
File Uploader	90%	10%	0	0	0
Notepad	80%	20%	0	0	0
Music Player	60%	20%	10%	10%	0

**Fig.8** Survey Result Table

Overall reviews showed that there is a need to improve upon the current applications and also to add more number of applications. The users also questioned the security services used to protect their data. A graphical representation of the overall ratings for all categories is provided below. The evaluation of our system was based on user satisfaction and application usability.



**Figure 9:** Survey Result graph

### ADVANTAGES

- Cost: As long as your system can run a web browser, you can access the application offered by your service provider. You won't have to upgrade your system every few years to keep up with the technology requirements of new software.
- File Sharing: File sharing is more convenient because you're using one common interface to create, modify and store files. Anyone can work on a single file from just about any computer as long as they have an internet connection and a web browser.
- Collaboration: Multiple users can work on the same file saved on a particular network from anywhere instead of having to organize several copies of one file and incorporate all of the changes into a new version.

These computing systems sometimes can be accessed with a common Web browser, but other users might gain access through a unique client. Although Web interface based operating systems often are designed to mimic a traditional OS, they are primarily a user interface.

### DISADVANTAGES

A number of shortcomings threaten to slow the adoption of Web interface based OSs. For example, they face network bandwidth and latency constraints that traditional OSs doesn't encounter. They also require extra layers of

overhead including power demands and the additional processing of keyboard, mouse, audio, and video inputs because they operate via a browser that runs on top of the host machine's operating system.

- If Web interface based OS users choose to locally cache data they're working with, the browser would have access to the information. This could cause security concerns if a hacker compromises the browser.
- Web interface based OSs generally comes with a limited number of applications. And, the Burton Group's Creese said, the applications might not have features that users like in their traditional programs. Also, users may be too unfamiliar with the software to want to use it.
- Web interface based OS platforms are relatively immature and don't always offer as much functionality, such as native device-driver access, as traditional OSs. This has hurt the Web OS's adoption by corporations, according to Creese.
- Most users can't easily transfer data and settings between different Web interfaces based OSs because they don't use the same data formats, although some technically sophisticated users could manually do so. Proponents still must develop standards and tools to handle interoperability. Companies

such as G.ho.st and Sapotek are working on such projects.

Generally, users must be online to work with a Web OS, a major problem for those who never or rarely connect to the Internet or who lose access temporarily.

## CONCLUSION

Web interface based operating system is one of the functionality of distributed system which provides access to an operating system free from the location. It makes it possible to organize a large variety of services and applications provided both by the internet and a typical desktop into the user's space. The aim is to provide a remote desktop that stores the user information remotely rather than on the users hard drive and accessible from any gadget that supports a browser.

This project has outlined a framework that integrates Service Oriented Architectures (SOA) which assists in simple addition of services to the existing service directory and the Semantic File system which uses semantics for storage, indexing and retrieval of data.

Adoption of semantic web interface based operating system presents an opportunity to change the conceptual model of desktop computing. Moving from a traditional position where the desktop is largely tied to a specific computational device, a semantic desktop could exist as a broad, networked space defined relative to the user. The Semantic web interface based operating system tries to

provide a virtual environment and the typical functionalities of a desktop where users can interact with the system as in a typical desktop environment. Such an online portal can also provide the user with computational power limited only by the bandwidth of the internet.

### Further Extension

- Further research and implementation will include security mechanisms
- Providing a more personalized environment to the user
- Bringing in a notion of open source by giving the user a method to add his/her own services
- Providing a wider range of services and applications and making the system more suitable for mobile and palm devices.

### REFERENCES

- [1] A. M. Riad, Hamdy K. Elminir, Mohamed Abu ElSoud, Sahar. F. Sabbeh. SEWOS: A FRAMEWORK FOR SEMANTIC WEB OPER-ATING SYSTEM. International Journal of Electrical & Computer Sciences IJECS IJENS Vol: 10 No: 01
- [2] M. P. Papazoglou, "Service Oriented Computing: Concepts, Characteristics and Directions", Proceedings of the Fourth International Conference on Web Information Systems Engineering,
- [3] <http://php.net/manual/en/language.references.php>: PHP manual
- [4] JavaScript: The Missing Manual; David Sawyer McFarland July 2008: First Edition
- [5] JavaScript For Dummies, 4th Edition Published by Wiley Publishing, Inc. Published by Wiley Publishing, Inc., Indianapolis, Indiana

- [6] PHP 5 For Dummies Published by Wiley Publishing, Inc. [www.wiley.com](http://www.wiley.com)
- [7] PHP & MySQL For Dummies, 2nd Edition Published by Wiley Publishing, Inc.
- [8] <http://www.peachpit.com/guides/guide.aspx?g=ash> : 2012 Pearson Education, Peachpit Press