



## Research Note

# Backend Deployment of User Interface for PSS

Riddhish Deotale, Dr. Arnab Das, Shridhar Prabhuraman

Maritime Research Centre

### Background

The Graphical User Interface or GUI uses icons, menus and other visual indicator (graphics) representations to display information and related user controls, unlike text-based interfaces, where data and commands are in text. GUI makes computer operation more intuitive, and thus makes the interaction of users with the software easy. GUIs generally provide users with immediate, visual feedback about the effect of each action. GUI allows multiple programs and/or instances to be displayed simultaneously. With ongoing utility in the application user can monitor the progress of the desired function, and also can make changes immediately instead of waiting for the ongoing process termination.

GUIs were introduced in reaction to the perceived steep learning curve of command-line interface (CLI), which requires commands to be typed on the keyboard. The major benefit of a GUI is that systems using one are accessible to people of all levels of knowledge, from an absolute beginner to an advanced developer or other tech-savvy individuals. They make it simple for anyone to open menus, move files, launch programs or search the internet without having to tell the computer via the command line to carry out a function[1].

### Domain Involved

A **user interface**, also called a "UI" or simply an "interface," is the means in which a person controls a software application or hardware device[2]. The user interface (UI), in the industrial design field of human-computer interaction, is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end[3]. A good

user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way[4].

In the beginning of digital computing, all computers were programmed, meaning that *all* user interfaces were programming interfaces. These were called Batch Monitors. The experience of using these early machines was not pointing, clicking, and typing, but rather punching cards to encode instructions for computers to execute. Once the cards were punched, one would drop them in a job queue and wait for the computer to go through the punches, then a printout containing the final result or an error message would be out. This interaction paradigm captured the early visions of computing as essentially calculation automation, freeing humans from the burden of arithmetic. The turnaround time for a single job often spanned entire days. If one were very lucky, it might be hours

**Command-line interfaces** (CLIs) evolved from batch monitors connected to the system console. Their interaction model was a series of request-response transactions, with requests expressed as textual commands in a specialized vocabulary. Latency was far lower than for batch systems, dropping from days or hours to seconds. And this also made the process Real Time, i.e. you can abort or edit the process when it is being executed. GUIs were introduced in the 1970s in reaction to the perceived steep learning curve of CLIs.

Before the start of GUI, there was CLI (Command Line Interface). At that time, no one thought normal people could use a computer. But, now everyone owns a computer and has a basic knowledge of how to use it. That is what GUI achieved[5]. GUI design has made tremendous progress subsequent to the introduction of the Macintosh and Windows 95, and it is continuing to advance. This is the result of several factors, including advances in computer hardware (e.g., faster processors, more memory and higher resolution display devices), improvements in software and ever-greater demands by users and application developers.

Some Drawbacks of using GUI are:

- It takes more power for the system to function[5].
- It is slow compared to simple command-based Interfaces[5].
- It consumes more memory space[5].

Presently GUIs are being developed for a three dimensional interaction with users, the illusion of navigating through a three-dimensional space, and the commands the machine receives are also being done using the voice commands.

One of the most distinctive features in the use of GUI that was never achieved in CLIs was to have data from various sources could be mapped and displayed in real-time on the screens. But with present technology and devices, it is possible to map the data onto screens for easy visualisations. Raw data that are present in text format are converted as point markers on the map and then they are interpolated, Interpolation is a commonly used GIS technique to create continuous surface from discrete points. After that hues, and contours are added for easy visualization. A lot of real world phenomena are continuous - elevations, soils, temperatures etc. If we wanted to model these surfaces for analysis, it is impossible to take measurements throughout the surface,

and thus there exists mapping softwares that helps in mapping the data[6]. Some of the most popular mapping softwares are QGIS 1,2,3; gVSI; GRASS GIS; ArcGIS.

### **Challenges & Opportunities**

GUI may be simple for a consumer but not as simple for the programmers who have to design and implement each and every function and also apply abstraction so that the users will feel the advantages of GUI[5]. GUI for a softwares are released in one of the two forms almost every time, they being **Responsive Website** and **Platform Specific app**. Where the biggest advantage of Responsive Websites is that it can be accessed through any type of platform with any screen size, the advantage of Platform Specific app is that it can be used to be accessed offline as well as online.

#### **Responsive Websites:**

The term Responsive means that the application designed can adjust to any screen size it is to be used, various features gets minimized or maximized depending upon it. Example of Responsive Website will be Most widely used search engine Google.

#### **Platform Specific Application:**

A Platform Specific Application is designed to be used on a particular platform only, it cannot be used on any other platform on which it is not meant to be used. Example of Platform Specific Application would be the most widely used Messaging app Whatsapp, it is particularly designed to be used on smartphones, and as a result we cannot access it on Desktop PCs.

When we discuss these two each has many advantages and disadvantages over each other. The major factors influencing the selection are:

#### **Functionalities:**

Giving a lot of functionalities is sometimes perceived in terms of the software worth but in reality the more functionalities we provide less the user's speed becomes. So there needs to be a right amount of functionalities an application should have if we need user speed. Biggest and most successful example of this is Google search engine, even when it has the least number of functionalities provided on its home page it is one of the most widely used search engines.

#### **Connectivity:**

Whether we want to keep our software completely offline or online or both the platform varies. Responsive Websites can only be accessed when it is connected to the internet and it should be connected to the internet the whole time you are using the application, while Platform Specific Application can be made both for online as well as offline mode.

## **Security:**

How secure we want the intellectual resources provided in our application to be also influences the selection of what way we want to develop our app. Platform Specific app are in general more secure than the Responsive Websites.

## **Mapping:**

There are many platforms that provide mapping features to other applications, the majority of them are paid software, but the most widely used one QGIS is an open source software, it is packed with many features making it slightly complicated to use. There are very few resources available on the internet related to mapping that makes it very challenging to use.

## **Integration of Maps with Software:**

You need an extensive knowledge of the programming language that the mapping software supports and in which you wish to design your application. QGIS is designed in C++ but supports python, Python is generally supported by almost all mapping softwares.

## **Future Research Scopes**

### **Development of Software to map data from Passive Sonars**

The Mapping potential provided by various mapping softwares can be used to map Maritime Raw Data for easy visualization we extract from Passive Sonars to Simulate a map that can show the density of the mapping fields like: Ambient Noise Mapping of low frequency generated due to shipping, Effective Detection, Vulnerability Assessment & Effective Deployment. Using the mapping feature many other maps and graph related work can be mapped.

### **Mapping a virtual 3D Map**

One of the most interesting areas of exploration is GUIs that provide the user with the illusion of navigating through a three-dimensional space[7].The Technology to map a 3D virtual Image/Hologram is being developed which is also being used on google earth to map data in a 3D view on a 2D Screen.

## **References**

[1] Hopping, Clare. "What Is a Graphical User Interface?" IT Pro , Dennis Publishing Ltd., May 2018, p. n/a.

[2] What is a hardware user interface? | AnswersDrive.  
<https://answersdrive.com/what-is-a-hardware-user-interface-465474>

[3] User interface - Wikipedia. [https://en.wikipedia.org/wiki/UI\\_designer](https://en.wikipedia.org/wiki/UI_designer)

[4] User Interface Definition. [https://techterms.com/definition/user\\_interface](https://techterms.com/definition/user_interface)

[5] What is GUI? | How It Works? | Need & Uses with Examples .... <https://www.educba.com/what-is-gui/>

[6] Interpolating Point Data — QGIS Tutorials and Tips.

[http://www.qgistutorials.com/en/docs/interpolating\\_point\\_data.html](http://www.qgistutorials.com/en/docs/interpolating_point_data.html)

[7] "GUI Definition - The Linux Information Project." 1 Oct. 2004, <http://www.linfo.org/gui.html>. Accessed 16 Jun. 2020.