CHAPTER 6
MINI 7 OPERATING SYSTEM

6.1 PREAMBLE

This chapter introduces a customized operating system called MINI 7 (derived from Windows 7 Operating system) to analyze the performance of innovative patterns. The results are presented and discussed by means of graphs and charts. The survey also has been made among stakeholders by distributing the questionnaire and the results are presented.

6.2 CASE STUDY 2: MINI 7 OPERATING SYSTEM

The performance of innovative patterns is supported by another case study using our customized operating system called Mini 7. Mini 7 is derived from WINDOWS 7 the latest version of Microsoft Windows. The original size is approximately 3 GB, by stripping away features on demand basis using a tool called V-Lite.

This process serves as a clear illustration of the application of innovative patterns on system software architecture in an advantageous manner.

6.2.1 Approach

This is straightforward. Features are selectively removed from Windows 7 on an on-demand basis so that we have a customized operating
system suitable for specific needs and environments (gaming, development, business etc). The result obtained should a significant increase in the quality criteria of the developed system.

The components removed are listed below:

Deployment Image Servicing and Management (DISM) and the following packages were also removed: Hyper V Common Drivers, Hyper V Guest Integration Drivers, Anytime Upgrade Results, Backup, Business Scanning Feature, Clips In The Library, Client Wired Network Drivers, Common ModemDrivers, Printer Drivers, Tuner Drivers, Win Ocr, Gadget Platform, Group Policy Client Extensions, IIS Web Server Add On, IIS Web Server Add On 2, Indexing Service, IE Troubleshooters, Links, LocalPack AU/CA/GB/ZA/US, Mobile PC Client Sensors, Optical Media Disc, PeerDist Client, Printing XPS Services, RasCMAK, RecDisc SDP, Sample Content Ringtones, Secure Startup, ShareMedia Control Panel, Shell HomeGroup, Shell Invoice Games, Sidebar Killbits SDP, Snipping Tool, Sticky Notes, Terminal Services Command Line Tools, Terminal Services Misc Redirection, Terminal Services Publishing WMI Provider, Terminal Services Remote Applications Client, Terminal Services USB Redirector, Terminal Services WMI Provider, Windows Media Player Troubleshooters, WMP Network Sharing Service, XPS Foundation Client, and Networking MPSSVC Rules Ultimate Edition.

- Windows\System32\DriverStore\FileRepository
- Windows\winsxs

6.2.2 Accessories

Accessibility, Inbox Games, Premium Inbox Games, Mobility Center, Speech Support, Windows Sidebar and Gadgets.
6.2.3 Drivers

Diva Server, Modems, Scanners, TV Tuners.

6.2.4 Languages

Except English and the regional language all other languages can be removed to reduce the overall size of the developed component.

6.2.5 Multimedia

Media Center, Music and Video samples, Sample Pictures, Screensavers, Sideshow, Sound Recorder, Windows Media Codes, and Windows Media Player.

6.2.6 Network


The size of the resulting product was a 699 MB which is more compact than the original product size which is 3 GB needing a DVD to hold its content. The next feature that we considered was the installation time of the developed software; the time taken for installing the mini 7 was 10 to 15 minutes vs. the time taken for installing Windows 7 that was about 30 min; thereby saving half the time used for installation. The next feature of the mini 7 was the space required to install the product. The size of the windows folder in mini 7 was 2.46 vs. the size of the Windows folder of Windows 7 was 8 GB. The final criterion that we considered was the number of services
running. The number of services running in Windows 7 was 45 needing 800 MB of RAM where as the mini 7 has only 24 processes running, consuming only about 300 MB of RAM, thereby increasing the availability of the RAM which, in turn, increases the system performance.

From the above data we can see that using innovative patterns there is a significant improvement in the performance and other quality criteria. Now, let us look at the ways in which each pattern is applied to the operating system.

6.3 SUBTRACTION PATTERN

The subtraction pattern was applied to the development of the operating system by removing all the unwanted components. After applying the subtraction pattern there was a significant improvement in the various quality criteria which will be discussed in this chapter.

While applying the subtraction pattern, we delete the services and components that are not required by the end product, like the File Repository in the system32 folder which reduces up to 550 MB of the total product size. By applying this subtraction pattern, we strip down the size of the operating system from approx 3 GB to about 699 MB, thereby reducing the development cost of the product.

6.4 DIVISION PATTERN AND TASK UNIFICATION PATTERN

The division and task unification patterns are applied to the operating system by applying the concept of multi-threading. In multi-threading we split the process into small threads which is the application of the division pattern. We also create threads from other processes that are
trying to run in the system, thereby improving the overall throughput of the processor which is achieved by the application of the task unification pattern.

6.5 ATTRIBUTE DEPENDENCY CHANGE PATTERN

The next criterion that we considered is the attribute dependency change. In Windows 7 we achieve this by setting an operating system from all kinds of users. For example, a home user gets either the home basic of a home premium like windows 7 ultimate. The idea here is, we tend to attain user satisfaction by applying the attribute dependency change pattern. Now, let us look at all the quality criteria that are associated with the application of these patterns in detail.

6.6 RESULTS AND DISCUSSION

Here we present the results of applying the innovative pattern approach for deriving an optimized version of Windows 7. This clearly brings out the benefits of the approach. The Modified is called mini 7 was analysed in terms of product size, function points, services and usability. The results are presented below in detail using various parameters like size, function points, services and usability.

6.6.1 Product Size

The first criterion that we consider is the size of the product. Consider the whole product as a large component which is split into smaller components. In this, we will look at some of the components that are used to build the operating system and how we are going to achieve a lightweight operating system by using the innovative patterns. The main idea here is that the size of the product also derives the cost of building the product.
From the above we can see that there is a change in the approach that we have used in developing the product. By removing the unwanted features we were able to reduce the size of the product from 3GB to 699MB.

The graph plotted below shows the relationship between the size of the component (in units of MB on y axis) and name of various component (x axis) and presents a comparison between the ATAM and innovative pattern approach as shown in Figure 6.1(a) and (b).

![Graph](image)

**Figure 6.1 Size**

where, AT - Aero Theme, WU - Windows Update, FR – File Repository
6.6.2 Function Points

The function points of a product are a unit to measure the business functionality of a product. In this example, we measure the function points with the size of the component; that is; for every 100MB we get a 1000 Function points. The function point helps us to derive the cost and the complexity of a particular product. Using the function points we can also derive the time required for developing the product.

The graph plotted below shows the relationship between function points (in units of size of the each service on y axis) and name of various component (x axis) and presents a comparison between the ATAM and innovative pattern approach as shown in Figure 6.2 (a) and (b).

![Graph of Function Points](image-url)

(a)

(b)

**Figure 6.2 Function points**

where, AT - Aero Theme, WU - Windows Update, FR – File Repository
6.6.3 Services

Here we are going to talk about the various services that are running in the operating system. As we have discussed, there are many services that are not required by the operating system; by removing these services we can obtain optimum working efficiency of the product.

Here we have listed out some services that are running in windows 7 and we have drawn a graph that makes a comparison between windows 7 and mini 7. The graph plotted below shows the relationship between services(in units of size of the each service in MB on y axis) and name of various component (x axis) and presents a comparison between the ATAM and innovative pattern approach as shown in Figure 6.3 (a) and (b).

![Figure 6.3 Services](image-url)
6.6.4 Usability

The final criterion that we considered was the usability of the product. The usability of the product depends mainly on the hardware compatibility of the software.

The idea is simple, as Windows 7 needs 700MB RAM to run its services; it needs at least 1GB RAM as its minimum requirement and it recommends at least 2GB RAM for its optimum performance.

But, in mini 7, it is a different story, as mini 7 needs only 300MB RAM to run all its services; so, the minimum requirement for mini 7 to run drops to 512MB, thereby allowing the product in systems of lower configuration. Finally, to make the mini 7 with optimum performance we need only 1GB RAM, thereby making the product usable to a wider range of users.

From the graph and charts we can see a significant increase in the usability of the product by applying the innovative patterns.

\[
\text{Usability} = \frac{\text{Number Of task that can be completed}}{\text{Amount of available memory}} \times 100
\]

The graph plotted below shows the relationship between usability (in units of percentage on y axis) and size of the RAM (x axis) and presents a comparison between the ATAM and innovative pattern approach as shown in Figures 6.4 (a) and (b).
The above results clearly show the overall performance of innovative pattern approach is definitely better than ATAM approach. The few more parameters like response time and turn around time is discussed in the next section.
6.7 RESPONSE TIME AND TURN AROUND TIME

The response time of an operating system is the interval between the request time and the first response to the request. The turn-around time of the operating system is the time taken from the time of submission of the job to the time of the completion of the job. By applying the subtraction pattern we see that there is a significant increase in these criteria. This is because as the footprint of the operating system reduces, the availability of the RAM increases which, in turn, increases the response time of the system.

The case study was analyzed and the effects of applying patterns were studied. This suggests that the above patterns are universally applicable.

A survey analysis also has been taken and discussed in the subsequent section.

6.8 SURVEY ANALYSIS

A survey has also been made by setting a questionnaire that was circulated among the stake-holders to predict the performance analysis of innovative patterns. The stakeholders were asked to circle the letter corresponding to their answer to the questions. If they hesitated between options, they were asked to choose the most conservative choice. If they have a doubt, they were asked to choose option D. It is shown in Figure 6.5.
Figure 6.5 Sample questionnaire used in the survey

The options and the questionnaire are listed below.

6.8.1 Questionnaire

Performance

1. The latency encountered while doing the transaction was
   A) Very low B) High but tolerable C) Unacceptably high
   (D) No Idea

Throughput

2. The number of normalized transactions that can be completed at peak load was
   A) Acceptable enough B) Barely acceptable C) Needs to be improved D) No Idea.
Usability-Proficiency training

3. The time you had to spend to gain proficiency with the nightingale software was
   A) Very short  B) Took a reasonable amount  C) Took too long and felt that it lacks user friendliness  D) No Idea

4. The context sensitive help provided with the software was
   A) Very useful  B) Not of much use  C) Useless  D) No Idea

Configurability

5. The time incurred in incorporating configuration changes was
   A) Acceptably low  B) Barely acceptable  C) Unacceptable  D) No Idea

Maintainability

6. The frequency of the occurrences of bugs was
   A) Too frequent  B) Frequent but tolerable  C) Rare  D) No Idea

7. Compatibility with the version change of components was
   A) Excellent  B) Satisfactory  C) Unsatisfactory  D) No Idea

Extensibility

8. Amount of effort in adding a new feature was
   A) Easy  B) Barely acceptable  C) Unacceptable  D) No Idea

Security

9. The security aspects of the system like confidentiality, intrusion resistance, are
   A) Excellent  B) Satisfactory  C) Unsatisfactory  D) No Idea
Availability

10. The system downtime is
   A) Negligible B) Annoying C) Unacceptable D) No Idea

11. The scalability of the system is
   A) Excellent B) Barely acceptable C) Unacceptable D) No Idea

Modularity

12. The degree of flexibility while replacing/ modifying components is
   A) Excellent B) Barely acceptable C) Unacceptable D) No Idea

The above questionnaire was used to collect feedback on the salient attributes of the software architecture. The subjects were divided into two groups. One group was given a software product in which patterns had not been applied (size =50).

Another group was given the software product derived after applying innovative patterns (size =50).

It was observed that higher ratings were received from the second group who were presented with software derived after applying innovative patterns. It was also observed that the ratings revived were in conformity with the graphs plotted above.
6.9 SUMMARY AND CONCLUSION

The results and discussions prove the influence of innovative patterns in enhancing performance of the evaluation of software architectures. The case study on Mini 7 illustrates how innovative patterns can be applied for customer driven optimisations.

A survey was conducted to assess the usability of the optimized product. Most of the stakeholders have given a positive ranking. From the above results and discussions it has been found that the stakeholders are very comfortable in applying innovative patterns for their problems rather than ATAM or any other architecture analysis techniques.