**SWE3005**

Principles of Design Patterns

**Submitted by**

**Tippireddypalli Thanusma**

**(21MIS7092)**

***To***

***DR. B.V.GOKULNATH***



**VIT-AP UNIVERSITY , AMARAVATHI**

**ANDHRA PRADESH**

**LAB – 04**

**Question :**

**How to support multiple look and feel standards.**

**Solution :-**

**Lexi User interface :**

The Lexi user interface is a sleek and intuitive platform designed to streamline the user experience in text editing and document creation. It features a minimalist design with a focus on clarity and efficiency, allowing users to concentrate on their content without distractions. Lexi offers a range of powerful tools and functionalities, including rich text formatting, real-time collaboration, version control, and seamless integration with cloud storage services. Its responsive design ensures smooth performance across various devices, from desktop computers to tablets and smartphones. With its user-friendly interface and robust feature set, Lexi empowers users to create and edit documents with ease, making it an invaluable tool for professionals, students, and creatives alike.

**The seven design problems of Lexi :**

* Document structure
* Formatting
* Embellishing the user interface
* Supporting multiple look-and-feel standards.
* Supporting multiple window systems.
* User operations.
* Spelling checking and hyphenation.

## Supporting Multiple Look-and-Feel Standards :

## In order to support multiple look-and-feel standards and achieve portability across different platforms without requiring a major overhaul, the design of Lexi's user interface needs to abstract object creation. This abstraction involves separating the process of creating user interface elements (glyphs) from the specific implementation details of each look-and-feel standard. By doing so, Lexi can dynamically instantiate the appropriate glyph classes based on the selected look-and-feel standard, enabling flexibility and ease of porting to new platforms.

## The proposed design approach involves two key components:

## Abstract Glyph Subclasses.

## Concrete Subclasses for Each Standard.

## Lexi needs a way to determine the look-and-feel standard that's being targeted in order to create the appropriate widgets. Not only must we avoid making explicit constructor calls; we must also be able to replace an entire widget set easily. We can achieve both by abstracting the process of object creation. An example will illustrate what we mean.

## 

### **Abstract Factory Pattern:**

Factories and products are the key participants in the [Abstract Factory (87)](http://www.cs.unc.edu/~stotts/GOF/hires/pat3afs.htm) pattern. This pattern captures how to create families of related product objects without instantiating classes directly. It's most appropriate when the number and general kinds of product objects stay constant, and there are differences in specific product families. We choose between families by instantiating a particular concrete factory and using it consistently to create products thereafter. We can also swap entire families of products by replacing the concrete factory with an instance of a different one. The Abstract Factory pattern's emphasis on families of products distinguishes it from other creational patterns, which involve only one kind of product object.

**Summary:**

We've applied eight different patterns to Lexi's design:

1. [Composite (163)](http://www.cs.unc.edu/~stotts/GOF/hires/pat4cfs.htm" \t "_mainDisplayFrame) to represent the document's physical structure,

1. [Strategy (315)](http://www.cs.unc.edu/~stotts/GOF/hires/pat5ifs.htm" \t "_mainDisplayFrame) to allow different formatting algorithms,

1. [Decorator (175)](http://www.cs.unc.edu/~stotts/GOF/hires/pat4dfs.htm" \t "_mainDisplayFrame) for embellishing the user interface,

1. [Abstract Factory (87)](http://www.cs.unc.edu/~stotts/GOF/hires/pat3afs.htm" \t "_mainDisplayFrame) for supporting multiple look-and-feel standards,

1. [Bridge (151)](http://www.cs.unc.edu/~stotts/GOF/hires/pat4bfs.htm" \t "_mainDisplayFrame) to allow multiple windowing platforms,

1. [Command (233)](http://www.cs.unc.edu/~stotts/GOF/hires/pat5bfs.htm" \t "_mainDisplayFrame) for undoable user operations,

1. [Iterator (257)](http://www.cs.unc.edu/~stotts/GOF/hires/pat5dfs.htm" \t "_mainDisplayFrame) for accessing and traversing object structures, and

1. [Visitor (331)](http://www.cs.unc.edu/~stotts/GOF/hires/pat5kfs.htm" \t "_mainDisplayFrame) for allowing an open-ended number of analytical capabilities without complicating the document structure's implementation.

