**Resumen del Articulo** ***Patrones de implementación para incluir comportamientos proactivos escrito*** por: Mailyn Moreno, Altemán Carrasco, Alejandro Rósete y Martha D. Delgado.

Habla sobre incorporar la proactividad en el software ya que existe una gran dependencia de los sistemas computarizados y es mayor la delegación de tareas para ellos y los patrones se han utilizado con éxito para la reducción de tiempo de desarrollo y así funcionar de guía para resolver un problema típico y tener un resultado con mayor calidad.

También comenta sobre el paradigma orientado a objetos la cual encapsula estados y comportamientos y crean la comunicación para iniciar un caso de uso través de un mensaje o petición de un actor y el comportamiento le da respuesta a ese mensaje.

Define patrón como instrucciones para hacer algo, menciona Christoper Alexander “Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, entonces describen el núcleo de la solución para ese problema, de tal forma que se puede usar esa solución un millón de veces”, nos describe como son los patrones y su comportamiento , por su lado los patrones de diseño comunican clases y objetos para solucionar un problema específico, estos se conforman de 4 elementos , Nombre, Problema, Solución y Consecuencia , y se dividen en 3 diferentes tipos creacionales (Ayuda a hacer un sistema independiente de cómo son creados , compuestos y representados sus objetos), estructurales (Relacionan las clases y objetos de la forma que son compuestas para formar estructuras más grandes como por ejemplo la herencia) y de comportamiento (tiene la relación con los algoritmos y la asignación de responsabilidades entre objetos).

En el articulo se hace referencia al patrón observer el cual forma parte del tipo de comportamiento, en este las intenciones se conocen y se relacionan en un ambiente que se observa, soluciona problemas o situaciones como, cuando una abstracción tiene dos aspectos, uno en función de las demás. cuando hay un cambio en una entidad, y es necesario cambiar a los demás y no se sabe cuántas entidades más habría que cambiar. cuando una entidad debe ser capaz de notificar a otras entidades sin hacer suposiciones acerca de quiénes son estas entidades.

Para implementar el patrón se necesitan dos entidades: ”Observador” y “Observado”, el primero como su nombre lo indica observa al observado y recibe una notificación si el observado sufre algún cambio , los agentes que cumplirán con estos roles tendrán la capacidad de comunicarse y actuar autónomamente, pudiendo hasta cambiar su comportamiento dependiendo de las situaciones a las que se enfrenten, en el documento se muestran diagramas de clases del observer , diagramas de secuencia y una vista de despliegue.

Describe un ejemplo de donde se observa el comportamiento del patrón de diseño, El Observatorio Tecnológico el cual se divide de una capa cliente y una capa que representa el observatorio y tiene un agente de fuente de datos, los usuarios utilizan la interfaz gráfica para acceder a lo que quieren , esto sería el observado el cual al percibir un cambio envía una notificación al observatorio en este caso es el observador el cual recibe la notificación del cambio y con respecto a este hace un cambio o aplica un método para responder a esa notificación con lo planeado del programa.

Se concluyo que al aplicar los patrones de diseño se puede comprobar de forma satisfactoria un comportamiento proactivo beneficioso para el usuario y a su ves mejora el rendimiento del código y le da más orden y mejor comprensión para otros programadores.