PATRON OBSERVADOR – MEDICIÓN DE TEMPERATURAS POR CIUDAD

Arquitectura Software Práctica 1

Nicolas Pueyo Soria – 870959
Yago Torres García - 878417

1. Introducción

El objetivo de este documento es explicar la primera parte de la primera práctica de la asignatura “Arquitectura Software”, en la que se pide realizar una aplicación con libertad en tema y lenguaje de programación, que implemente el patron observador, así como alguna de las mejoras especificadas en el documento “Observer, Object Behavioral”.

2. Diagrama de clases y explicaciones

A continuación se muestran dos imágenes sacadas de la herramienta “Visual Páradigm 17.1” con la que se ha documentado la arquitectura del programa realizado para la práctica, un diagrama de clases y otro de secuencia.

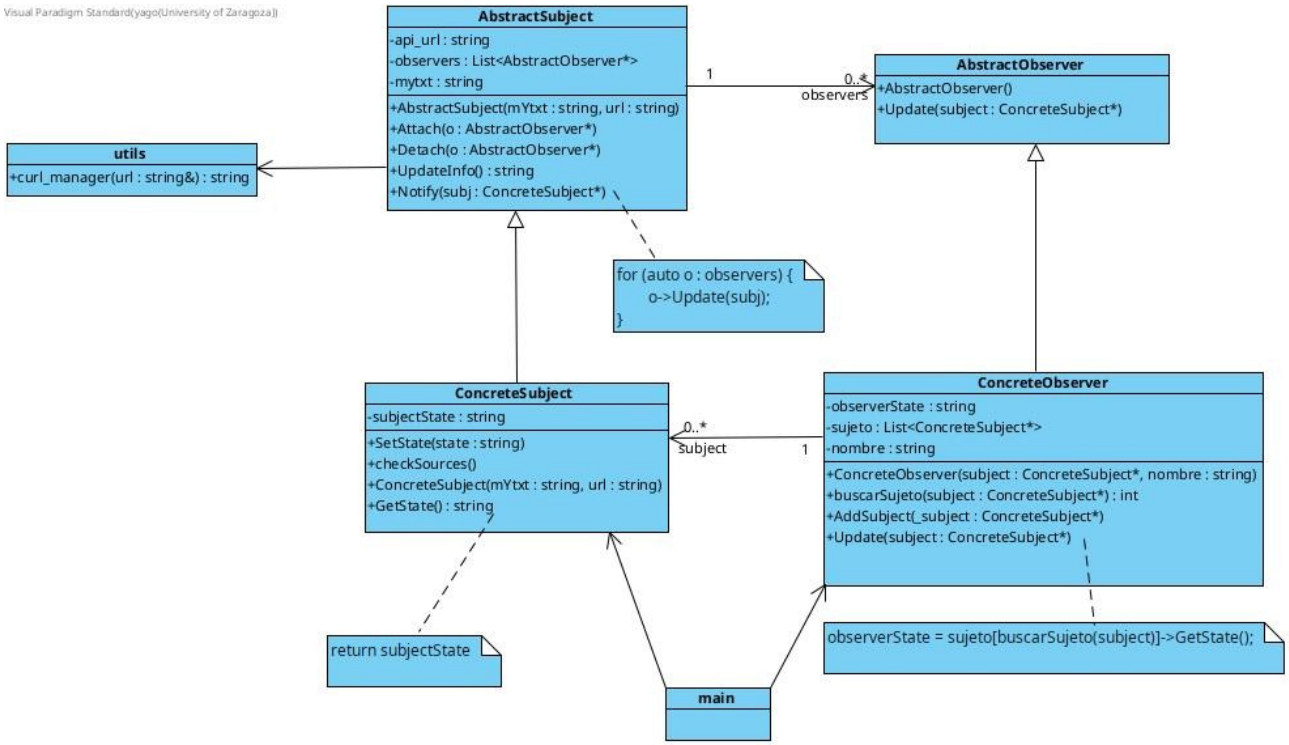

Figura 1 – Diagrama de clases

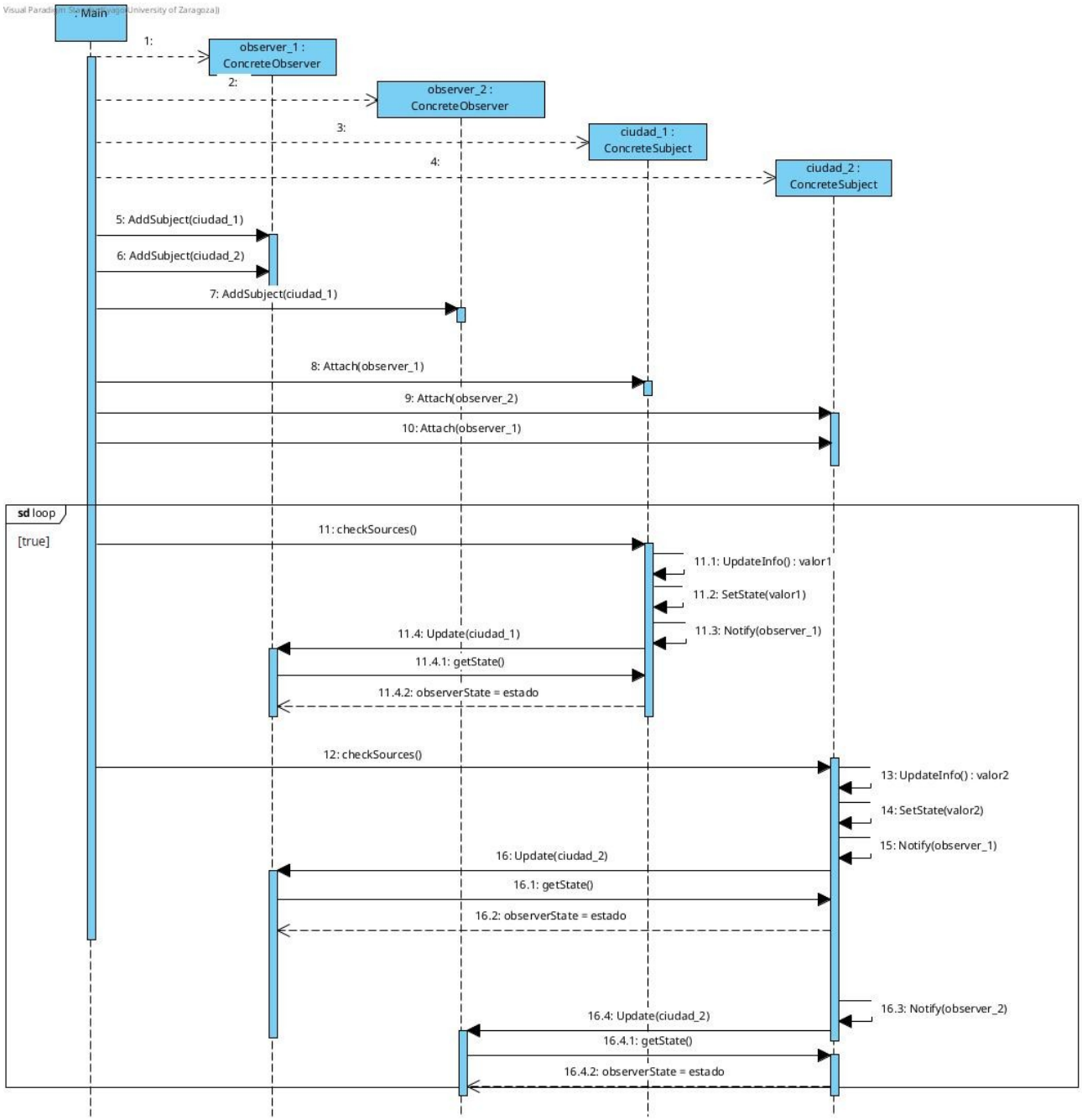

Figura 2 – Diagrama de secuencia

Viendo estos dos diagramas, se puede ver claramente como la implementación sigue la misma idea que la documentación del patrón en el documento.

En cuanto a las clases, se ve como contamos con equivalencias directas con las clases del patrón documentado, teniendo un sujeto abstracto y concreto, así como un observador abstracto y concreto. Estas clases tienen, además, las mismas funciones especificadas en el patrón. Estas funciones han sido modificadas parcialmente para conseguir algunas de las modificaciones que se especifican posteriormente en este documento.

Las relaciones también son análogas a las del patrón, heredando las funciones las dos clases concretas y añadiendo una condición a la relación entre los observadores concretos y los sujetos concretos, que permite que un observador concreto observe a varios sujetos.

La sección “Collaborations” del documento explica cual es la interacción al actualizar y luego recibir el estado de los observadores y sujetos. Se puede ver la similitud a la hora de realizarse en nuestra implementación especialmente en el diagrama de secuencia, en las secciones 16-16.2, que es la que propone el documento como ejemplo.

Con respecto a la adición de un protocolo que ayude a los observadores a descubrir partes del sujeto, debido a la naturaleza de nuestrá aplicación (simplemente observar la temperatura y su cambio en los sujetos ciudad) no tenemos “partes” del subject que queramos comprobar solas. Sin embargo, si se añadieran nuevas secciones (por ejemplo, precipitaciones en la ciudad) se podría gestionar esta actualización de solo la parte pasando como parámetro en la función *Update(subject : ConcreteSubject*)* que parte de la información ha sido modificada.

Varias de las mejoras propuestas en la sección de implementación proporcionado en Moodle han sido añadidas a nuestra implementación del patrón, que se explican a continuación:

- Observar más de un sujeto: Hemos considerado esencial la posibilidad de observar más de un sujeto en nuestro caso, para poder ver la temperatura de varias ciudades. En este caso, se ha seguido la lógica mencionada en el propio documento, pasarse a si mismo como parámetro en la llamada a update para que el observador sepa cual es el que llama, y no tenga que comprobar cambios en todos.

- Llamada a *Notify()* automática: Viendo la necesidad de avisar a los observers cuando se cambia el estado, y que tenemos una función *setState()* que se llama cada vez que se realiza un cambio, se considera la opción A, en la que la propia función *setState()* realizará la notificación a los observadores de que ha habido un cambio.

- Estado consistente previo a la notificación: Debido a realizar la llamada a *Notify()* de forma automática como se ha mencionado, se consigue que en el momento en el que se realice esta acción el valor de la temperatura es ya consistente (el último recibido).

- Pull model: En nuestro caso, hemos optado por realizar una implementación cercana a la “pull-model”, en la que la información pasada al observer con la notificación es mínima. Esto hace que la reusabilidad del código actual y las llamadas sea muy elevada, ya que es el observador el que se encargará de coger la información que necesite de el sujeto.