

En los test 01 y 02 hay código repetido. Cuando lo extrajeron crearon algo nuevo. Eso es algo que estaba en la realidad y no estaba representado en nuestro código, por eso teníamos código repetido. ¿Cuál es esa entidad de la realidad que crearon?

No creamos nada, sino que usamos una clase, Time, que ya estaba implementado en Smalltalk. Lo que hicimos es tomar aquella porción del dominio de la realidad que representa al tiempo en generar cierta acción. No tendría sentido generar una nueva abstracción sabiendo que ya hay una abstracción implementada en Smalltalk con el mismo objetivo de tomar el tiempo cuando se realiza una actividad.

¿Cuáles son las formas en que podemos representar entes de la realidad en Smalltalk que conocés? Es decir, ¿qué cosas del lenguaje Smalltalk puedo usar para representar entidades de la realidad?

Usando el 1° Paradigma en el Denotative podíamos representar una porción del dominio de la realidad con objetos particulares, también podemos tener objetos padres y sus hijos, donde se realiza esto cuando tenemos 2 porciones de la realidad que se comportan de una manera bastante similar. Por ejemplo, si pensamos en un Auto podríamos pensar directamente en un auto particular y de ello que se comporte como un objeto particular. O por ejemplo lo mismo que sucedía con el Humano Arthas, Orco Olgra y Mankrik, es decir, son dominios de la realidad específicos.

Por otro lado está el 2° Paradigma donde se pueden implementar las clases en el System Browser.

Lo bueno de este paradigma es que permite una mejor abstracción de la realidad, por ejemplo, mediante una jerarquización de clases con sus subclases podemos formular mejor el mundo de las 'ideas', es decir, tener conceptos, y el mundo sensible, que es acá cuando ya tenemos clases bastante particulares. Por ejemplo, Ser Vivo -> Vertebrado -> Mamífero -> Perro. Acá tenemos una jerarquización de clases, donde las primeras clases son abstractas, para ya ir pasando a algo más particular. Y si queremos generar representaciones de la realidad específicas ya podemos generar objetos como instancias de distintas clases.

¿Qué relación hay entre sacar código repetido (creando abstracciones) y la teoría del modelo/sistema (del paper de Naur)?

El paper de Naur habla de una Visión de Construcción de la Teoría (también llamado VCT) donde el programador tiene que tener una visión de como serán manejados ciertos aspectos de la realidad, o representados por un programa de la computadora.

Esto también se puede relacionar con el hecho de sacar código repetido, porque estamos generando abstracciones, es decir que el programador tiene manejados los aspectos del mundo real y sabe en qué generar ideas abstractas y en que no. Además de generar una mayor legibilidad. Es decir, el programador debe entender el mundo y saber qué cosas toma de él y que no, y en base a ello entender su dominio para saber en qué hacer abstracciones.