

Chalten - Completando un Modelo de Representación del Tiempo

Maximiliano Taborda
Mercap S.R.L. - Software Developer
m.taborda@mercapsoftware.com

La información temporal es crucial en varias aplicaciones informáticas. Existen diferentes tipos de entidades temporales tales como las que representan puntos en el tiempo o aquellas que definen medidas de tiempo. A lo largo de la historia de la informática se realizaron investigaciones en modelos temporales pero la comunidad Smalltalk no fue lo suficientemente beneficiada con ellas.

Smalltalk-80 provee las clases Date y Time para modelar el dominio del tiempo. Estas abstracciones no son suficientes cuando se deben programar observaciones complejas sobre el tiempo. ANSI Smalltalk agregó a estas las clases Duration y DateAndTime. Squeak amplió el modelo con las abstracciones de Timespan, Year, Month y Week. Este modelo carece de algunas abstracciones y no modela apropiadamente el problema del dominio. El modelo propuesto por Wilkinson y demás¹ solucionó varias carencias del modelo de Squeak y agregó simplicidad al trabajo del programador, pero solo se ocupó de abstraer cuestiones referentes al Calendario Gregoriano. Además al ser un paquete propietario de Mercap S.R.L. la comunidad Smalltalk no cuenta con estos beneficios.

Se propone para la presentación mostrar la investigación que se está realizando para plasmar en un modelo de objetos cómo los seres humanos medimos, nombramos y formalizamos el transcurso del tiempo con distintos calendarios. El objetivo de este modelo es permitir al desarrollador trabajar de manera simple y entendible con las entidades que forman parte de la realidad del dominio mencionado.

Palabras Claves: Calendario, Punto en el Tiempo, Fecha, Hora, Smalltalk, Objetos.

Introducción

El objetivo del trabajo es mostrar un modelo creado para representar distintas maneras y convenciones de medir el tiempo. El modelo obtenido es el resultado de la evolución que sufrió un modelo que, con algunas falencias y limitaciones, solo representaba ciertas características del Calendario Gregoriano, llegando, hasta el momento, a representar otros calendarios, tales como el Juliano, el Islámico y el Israelí, y que resuelve limitaciones existentes en el modelo original como por ejemplo la representación del infinito y del menos infinito, temporalmente hablando.

El modelo estudiado será presentado en vivo en la conferencia.

La presentación esta dirigida a todas aquellas personas que necesiten trabajar con objetos que representen las distintas entidades del tiempo.

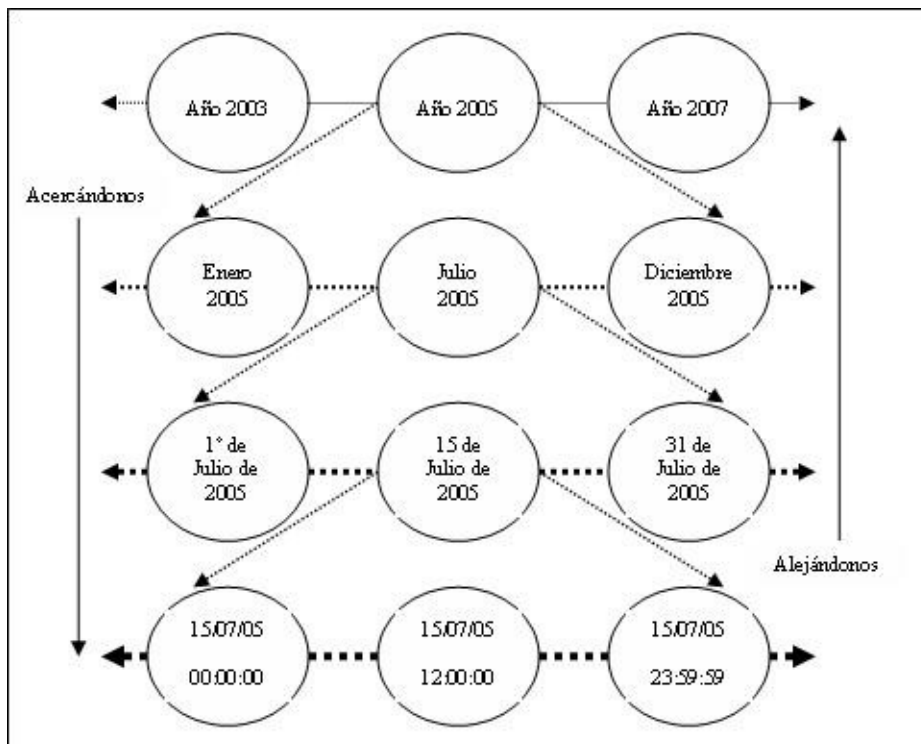
La presentación está preparada para ser realizada en el track de "Investigación y Educación"

¹ A New Object-Oriented Model of The Gregorian Calendar – Hernán Wilkinson, Máximo Prieto, Luciano Romeo

Desarrollo del Modelo

Muchas veces para encarar la solución a un problema complicado es conveniente buscar una metáfora que haga el problema más fácil de entender. El dominio del tiempo presenta ciertas complicaciones principalmente ocasionadas por la irregularidad del mismo, por lo tanto para facilitar el entendimiento del mismo, puede utilizarse la metáfora enunciada por Wilkinson y demás. En esta metáfora, las entidades temporales pueden verse como puntos pertenecientes a una recta que represente la línea del tiempo.

Al observar esa línea uno se puede acercar o alejar de los puntos que contiene. A medida que nos acercamos a la línea veremos los puntos mas pequeños (por ejemplo las fechas), y cuando nos alejamos se hacen visibles los puntos más grandes de la línea (por ejemplo los años). Se puede decir entonces que esa línea tiene diferentes escalas, o que esas líneas del tiempo de diferente escala pueden representar el paso del tiempo.



Acercándonos y alejándonos de la línea del tiempo vemos la misma en diferentes escalas

Por ejemplo, un año representa un punto en el tiempo pero con menor resolución que una fecha. Si vemos ese año más de cerca, se pueden observar nuevos puntos; esos puntos son los meses de ese año. De la misma manera, si tomamos uno de esos puntos (un mes de ese año) y nos acercamos podremos ver los puntos que representan las fechas de ese mes. Si seguimos acercándonos por los puntos que van apareciendo podremos ver la representación del tiempo en diferentes granularidades.

Lo mismo sucede en el camino inverso. Si tomamos una hora de una fecha y nos alejamos veremos un punto que representa esa fecha. Al mirar ese día desde un lugar más alejado podremos ver el mes al cual pertenece dicha fecha.

Basado en este principio, fue que originalmente se observaron y modelaron los siguientes conceptos, entre otros:

- Años: Modelados por medio de la clase GregorianYear. Esta clase representa los años como por ejemplo el "Año 2007".
- Meses de un Año: Modelados con la clase GregorianMonthOfYear. Esta clase representa

entidades tales como "Octubre del Año 2007".

- Fechas: Modelados a través de la clase `GregorianCalendar`. Representa cosas como "16/10/2007", lo que es el "16 de Octubre del Año 2007".
- Fechas Relativas: Modelados con la clase `RelativeGregorianCalendar`. Es usada para fechas que pueden cambiar dependiendo de diferentes eventos temporales, como ser días hábiles y no hábiles.
- Hora de un Fecha dada: Representado con la clase `GregorianCalendar`. Esta clase modela entidades como ser "16/10/2007 21:05:00", lo que es el "16 de Octubre del Año 2007 a las 21:05 hs."
- Días de un Mes: Modelado con la clase `GregorianCalendar`. Esta clase representa cosas como "16 de Octubre". Estos son días de un mes pero de ningún año en particular.
- Meses: Modelados con la clase `GregorianCalendar`. Estos meses son entidades como "Enero", "Febrero", "Marzo", etc.
- Días: Modelados con la clase `GregorianCalendar`. Estos son los días, como por ejemplo "Domingo", "Lunes", "Martes", etc.
- Hora de un Día: Modelado con la clase `TimeOfDay`. Esto representa la hora en un día, como por ejemplo "10 AM", "12 PM", "21:15:50".
- Segmentos de la Línea del Tiempo: Modelado con la clase `TimeSpan`, por ejemplo "10 días desde ahora".

Falencias encontradas:

- No se modeló el concepto de calendario.
- `JanuaryGregorianCalendar` y `FebruaryGregorianCalendar`; ¿es realmente necesario especificar estos dos meses? La razón de la primera es porque marca el inicio del año, y la segunda porque cambia su definición en los años bisiestos.
- Todo fue modelado como un punto en la línea del tiempo, y conceptos como el mes (`GregorianCalendar`) o el día (`GregorianCalendar`) son más bien entidades que podrían pensarse en un círculo más que en una recta.
- Si bien se trato de representar la irregularidad en la forma de contar el paso del tiempo, hay cuestiones que no se abarcan como el tener años con diferente cantidad de meses, por ejemplo, en el Calendario Israelí el año bisiesto presenta un mes más que el año no bisiesto.
- Se pueden especificar fechas relativas, pero no otras entidades relativas; o sea, el modelo permite representar que fecha será dentro de tres meses a partir de la fecha de hoy, pero no se puede hablar a nivel mes o año de manera relativa.

Con la presentación se intentará mostrar como se fueron solucionando estas y otras limitaciones encontradas al modelo original. El desarrollo de estas soluciones se llevo a cabo haciendo incapié en un buen modelo donde se refleje cada entidad de la realidad con un objeto que represente a la misma de una manera lo más fiel posible, para así lograr que el gap existente entre entidad de la realidad y representación de la misma en el modelo computacional sea lo más acotado posible.