

Sexta semana de trabajo [12 - 16 de Mayo]

Milton Inostroza Aguilera

19 Mayo de 2008

Resumen

Se modifica borrador de protocolo para mejor adaptación entre pyTOD y TOD:

- instanciación es agregado a la tabla de sucesos.
- Se modifica por completo el registro de return.

Se ha logrado solucionar los problemas que se tenían para ver el programa depurado desde la interfaz de TOD. Se agregó, para el buen funcionamiento de la interfaz de TOD, el suceso de instanciación.

Para registrar las funciones, se ha creado una clase especial en la base de datos TOD, la cual contendrá todas las funciones que se utilicen en el lado Python.

<i>INDICE</i>	2
---------------	---

Indice

1	Desarrollo	5
2	Depuraciones	5

Lista de Figuras

1	Pantalla principal de TOD	6
2	Vista de todos los eventos	7
3	Vista del control de flujo	7
4	Evento instanciación seleccionado	8
5	Clases, métodos y funciones	8
6	Métodos con variables locales	9
7	Clase para almacenar funciones	9

Lista de Tablas

1 Desarrollo

El problema que se tenía con los timestamp era básicamente que el frame y la llamada al método/función tenían timestamp diferentes. Se unificaron estos timestamps y se logró que la interfaz de TOD funcionara correctamente.

Las llamadas de métodos/funciones no mostraban el valor retornado por las mismas en la interfaz TOD. Se detectó que el problema era que la profundidad debía ser $\text{depth} + 1$ para el registro de return y set de variables locales/atributos.

Sigue siendo necesario la modificación de la máquina virtual de Python o la creación de un tipo de datos nativo especial que permita implementar el identificador único para ver como se comporta pyTOD con este tipo de objetos.

2 Depuraciones

Después de haber solucionado los problemas anteriormente descritos, ya podemos ver la interfaz gráfica funcionando de forma más completa.

Se ha depurado experimentalmente el siguiente trozo de código:

```
class clase1(Descriptor):  
  
    def __init__(self, y):  
        self.x = 1  
        self.c = 2  
        self.z = 3  
        return  
  
    def metodo(self, h, i, j, k):  
        self.casa = 1  
        k = i + j  
        return
```

TOD, nos muestra la siguiente información:

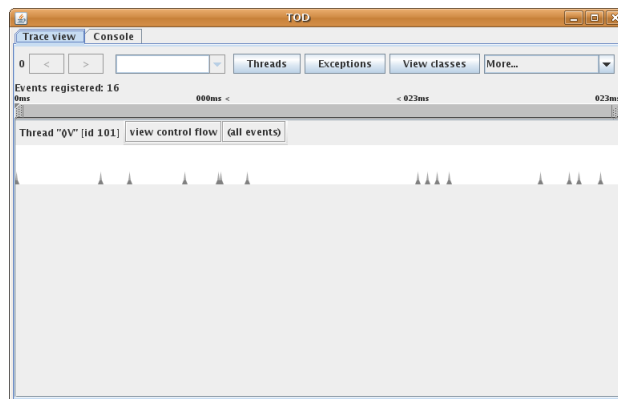


Figura 1: Pantalla principal de TOD

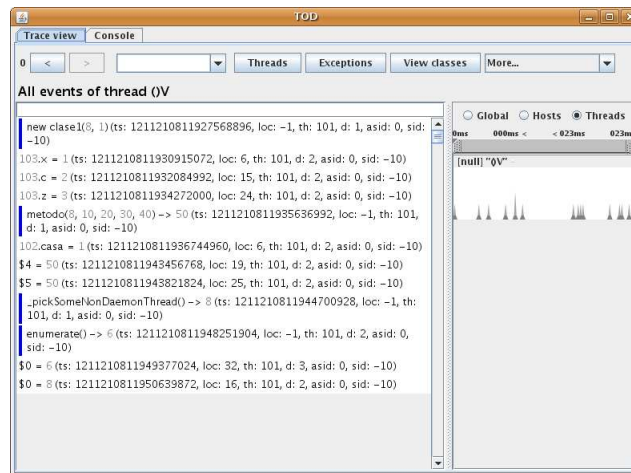


Figura 2: Vista de todos los eventos

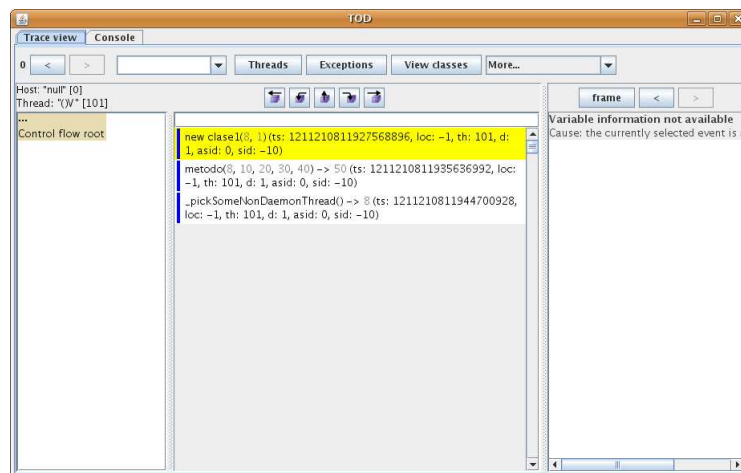


Figura 3: Vista del control de flujo

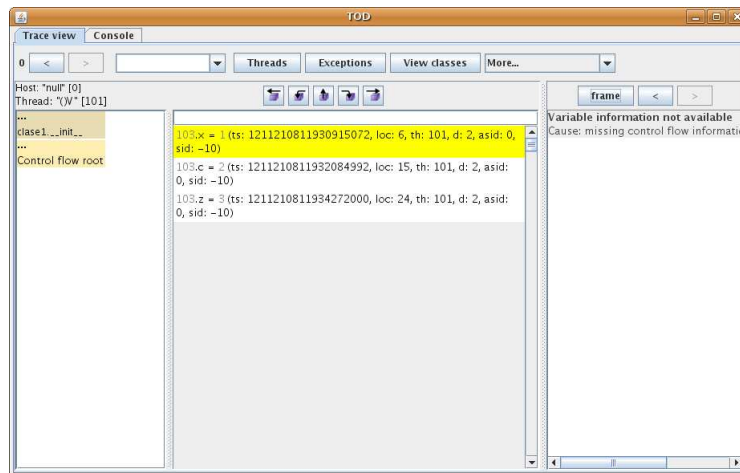


Figura 4: Evento instanciación seleccionado

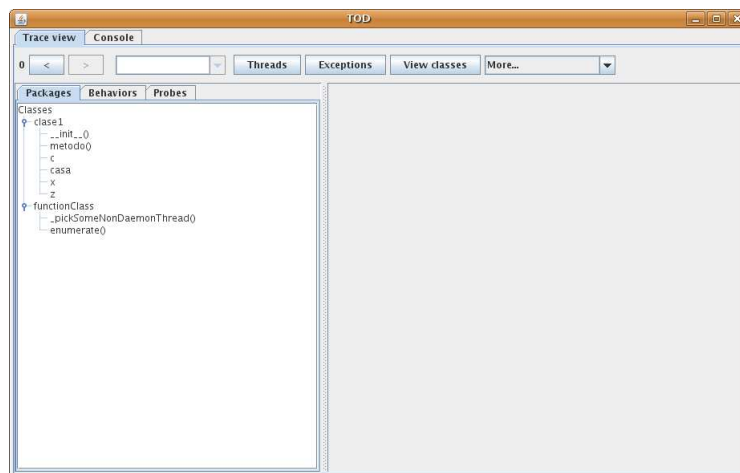


Figura 5: Clases, métodos y funciones

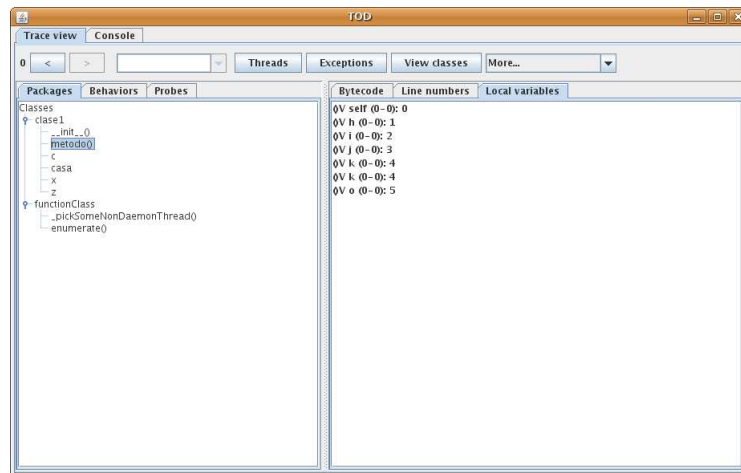


Figura 6: Métodos con variables locales

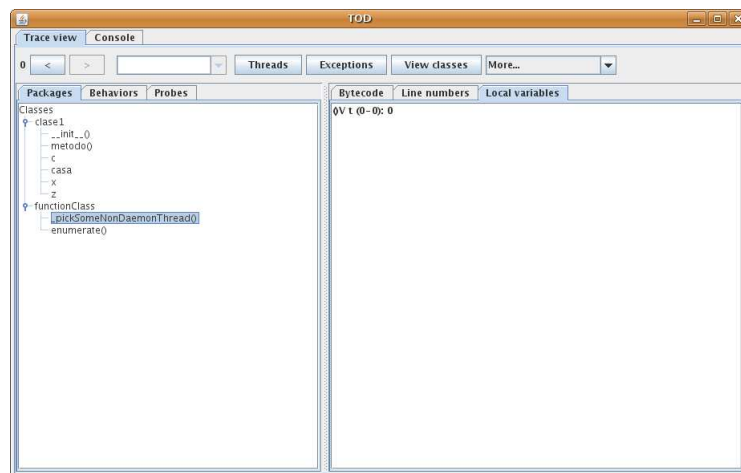


Figura 7: Clase para almacenar funciones

Es importante señalar que las funciones que aparecen registradas no son de nuestro código pero pertenecen a la implementación Python para el soporte de threads.

Bibliografía

- [1] <http://docs.python.org/lib/built-in-funcs.html>
- [2] <http://docs.python.org/ext/ext.html>