Homework 2: Reflection and Multiple Dispatch in Smalltalk

Due: Problems 1 and 2, September 24, 2002; problems 3 and 4, October 8, 2002.

This homework can either be done individually or in teams. Its purpose is to familiarize you with Smalltalk's reflection facilities and to help learn about other issues in OO language design.

Don't hesitate to contact the staff if you are not clear about what to do.

In this homework, you will adapt the "multiple dispatch as dispatch on tuples" approach, originated by Leavens and Millstein [1], to Smalltalk. To do this you will have to read their paper and then do the following.

1. (30 points) In a new category, `Tuple-Smalltalk`, add a class `Tuple`, which has indexed instance variables. Design this type to provide the necessary methods for working with dispatch on tuples, for example, we'd like to have access to the tuple of classes of the objects in a given tuple. Some of these methods will depend on what's below. Also design methods so that tuples can be used as a data structure (e.g., to pass multiple arguments and return multiple results).

2. (50 points) In the category, `Tuple-Smalltalk`, design other classes to hold the behavior that will be declared for tuples. For example, you might want something analogous to a method dictionary and other things found in the class `Behavior` or `ClassDescription`. It's worth studying those classes to see what can be adapted to our purposes.

3. (80 points) In a new category, `Tuple-Browser`, design something like a class browser for declaring and adding methods based on tuples. You should study the Smalltalk class `Browser` and its subclasses for ideas. This may involve modifying the Smalltalk compiler. Look in the category `System-Compiler` for more about the compiler that is part of Squeak.

4. (50 points) Using your browser, write several small example programs in Tuple Smalltalk to illustrate the language and its ideas. Test the individual pieces before putting everything together.

Hand in an overview of what you did to solve this problem (a short description) and printouts of the relevant Smalltalk code. Describe how you tested and show your test inputs (for example SUnit tests).

References

[1] Gary T. Leavens and Todd D. Millstein. Multiple dispatch as dispatch on tuples. In *OOPSLA '98 Conference Proceedings*, volume 33(10) of *ACM SIGPLAN Notices*, pages 374–387, October 1998.