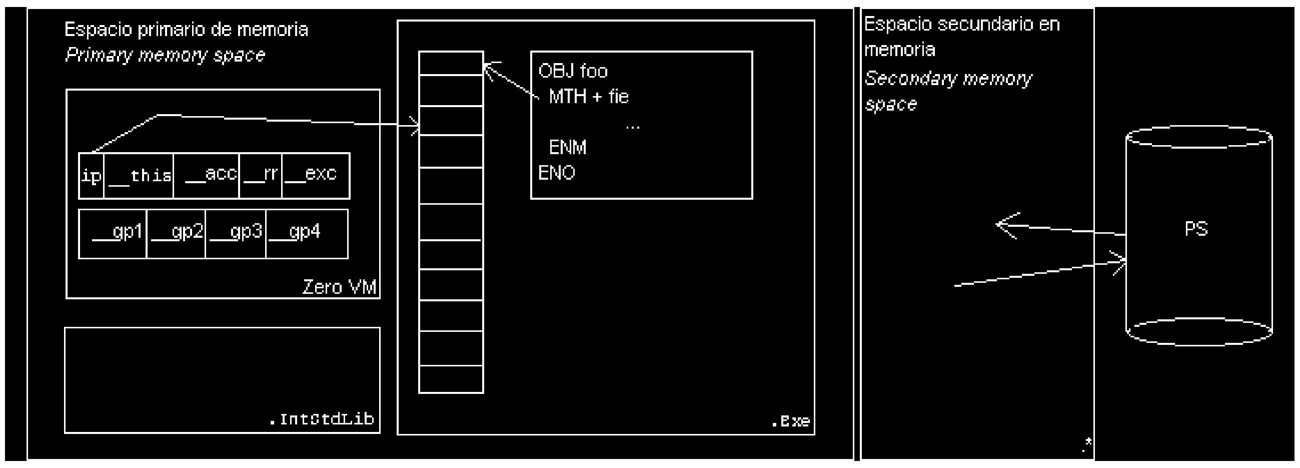
Zero: Towards an educational object-oriented programming environment

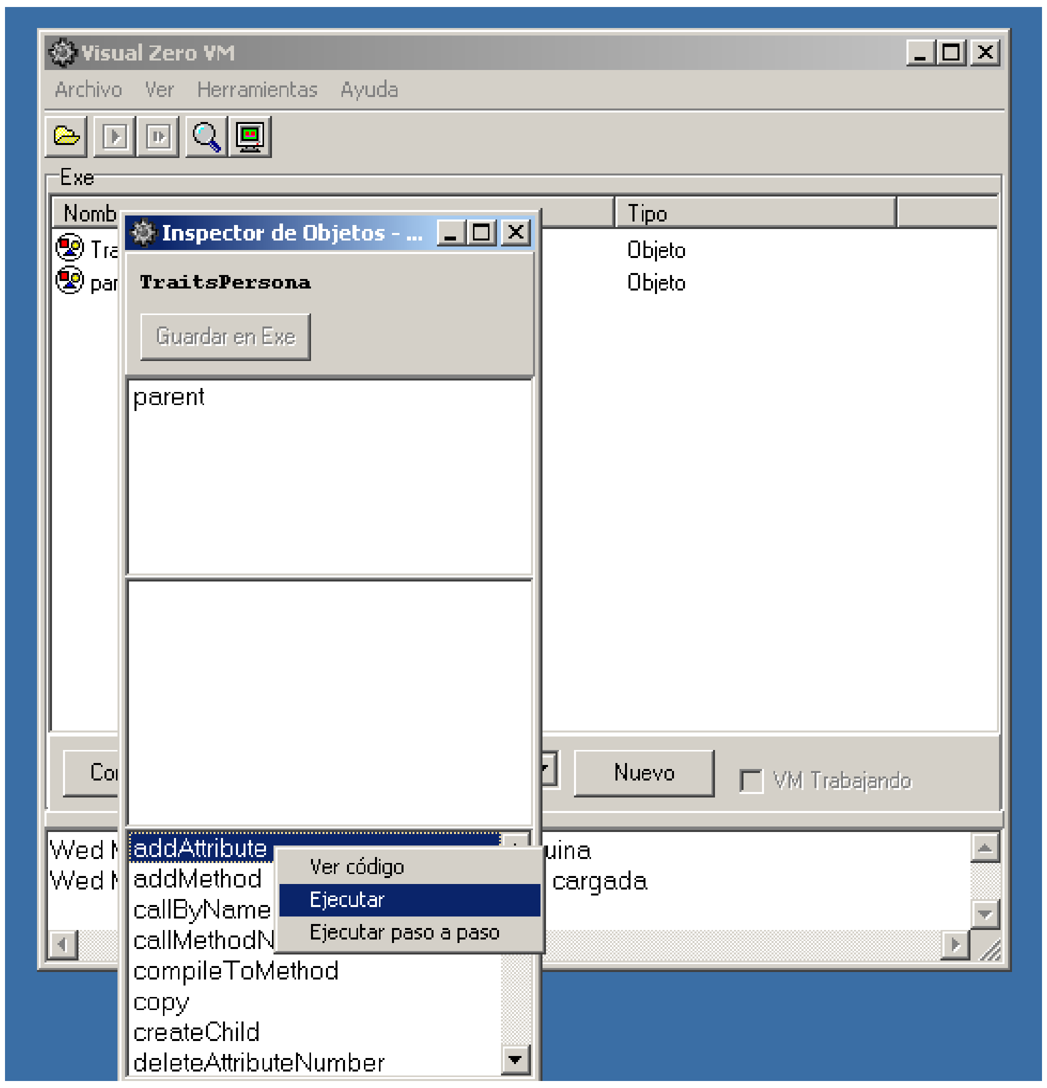
University of Vigo

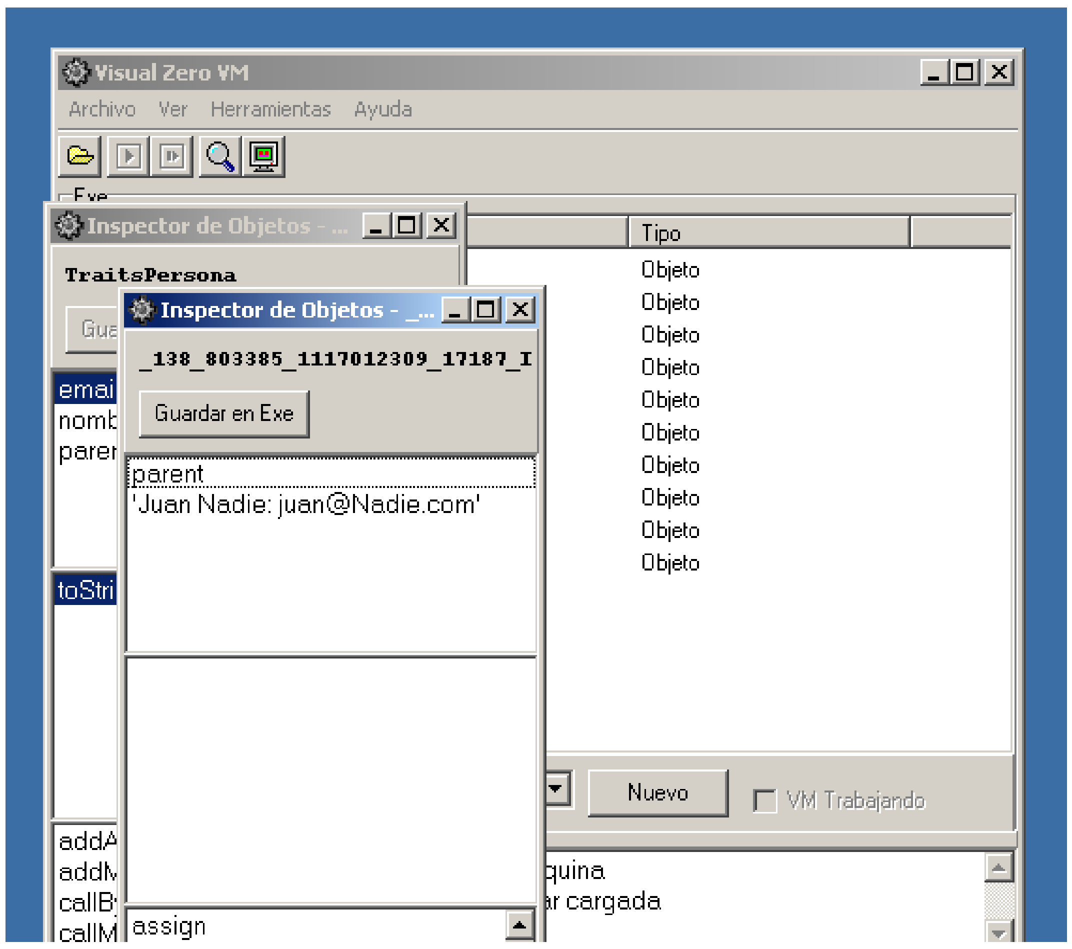


Zero is an ongoing research project held in the Computer Science department of the University of Vigo. Its main objective is to serve as a vehicle for learning object-oriented programming.

It currently consists of a virtual machine (its architecture is shown above), assembler and macroassembler, two high-level language compilers, and an educational programming environment. Although it is not still finished completely. it has been successfully employed in a number of different courses. Its main characteristics are (a) the support of prototype-based object orientation, which is a model of object-orientation that actually wraps the class-based model; (b) the future support for object persistence, which simplifies to the minimum all input/output issues, and (c) the support of multiple platforms, through portable bytecode. ***Visual Zero*** is an educational programming environment distributed with Zero. By using this program, students can learn quickly how to use an object calling its methods and inspecting its attributes. The system allows creating, deleting and modifying them as well.

In this screenshot we can see ***Visual Zero*** showing an *object explorer*. This tool, allows the user to manipulate an object in any possible way (multiple explorers can be opened at the same time, one for each object being inspected). The explorer is divided in three interactive lists: the first one shows the attributes present in the object, the second one the methods defined by the object, and the last one the inherited methods. In the picture, the method `addAttribute` is about to be executed. It is only needed to mark the method and select *Ejecutar* (execute) from the contextual menu.

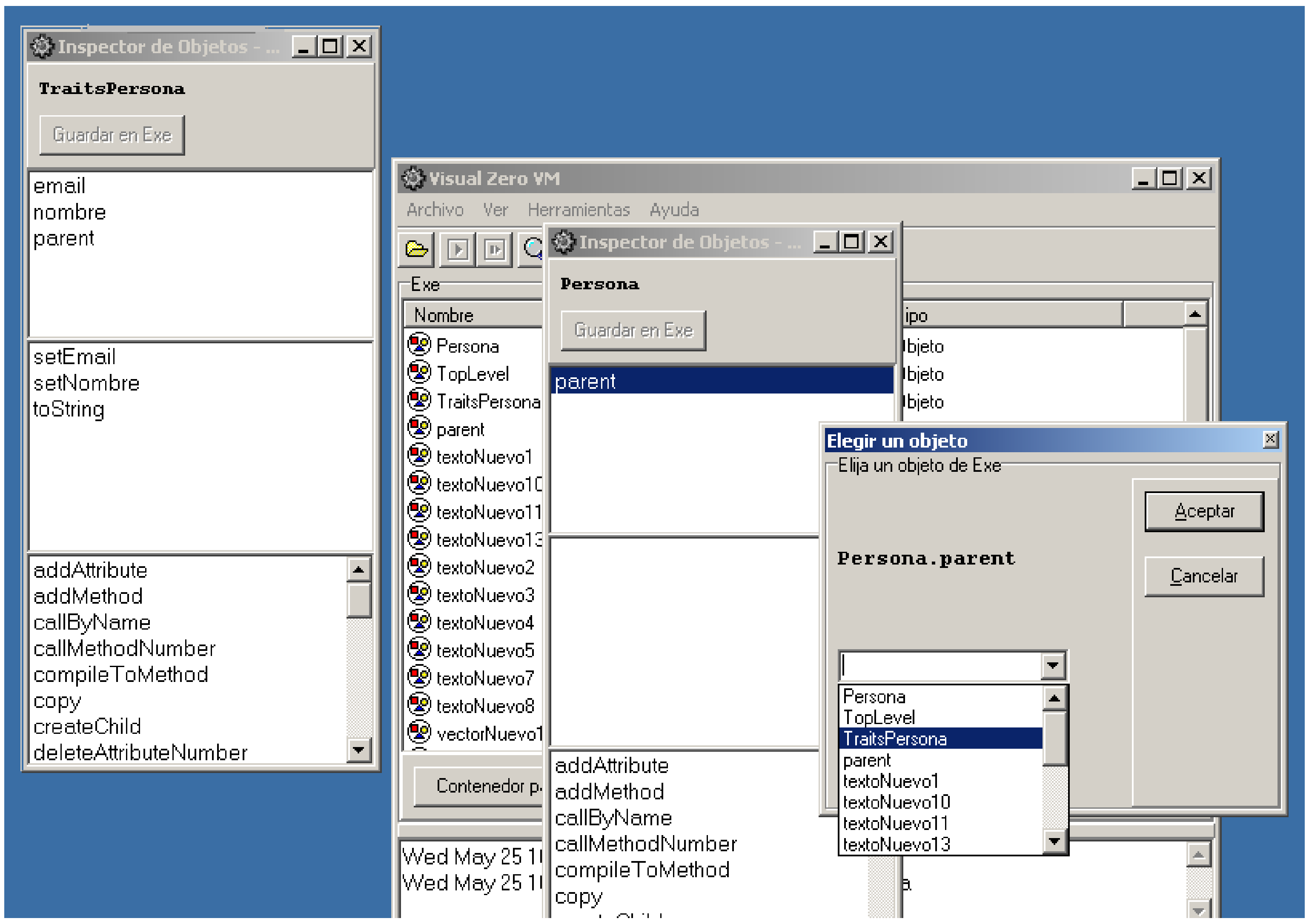




```
parent
'Juan Nadie: juan@Nadie.com'
```

The picture on the left shows the result of the execution of a simpler method: `toString`. This method returns the textual description of an object as text, in this case, the description of a person (*TraitsPersona*), which is going to be the prototype for all person objects in the system. The commented method returns the name of the person and his or her e.mail, separated by a colon.
The result of the execution is a **String** object, and therefore it is presented in an *object explorer*, as expected.

The screenshot below shows how to change a given attribute of an object: a pull-down list is shown, and the user just chooses the object he or she wants. The attribute that is being changed is important because it is the parent attribute of the object. This is called dynamic inheritance, and it is supported by the system.



J. Baltasar García Perez-Schofield IMO Group
http://webs.uvigo.es/jbgarcia/ jbgarcia@uvigo.es
SI1 – *Department of Computer Science*
University of Vigo