

## **Moliere: a Visual Environment using the Theatre Metaphor**

Isabelle BORNE  
Universite René Descartes  
UFR de Mathématiques et Informatique  
45, rue des Saints-Pères  
F- 75006 Paris  
E-mail: borne.laforia.ibp.fr

Metaphors can facilitate access to a system by novice programmers. Experiments have shown that object-oriented programming poses problems to students at both syntactic and conceptual levels. In educational settings, several object-oriented systems, based on new metaphors and on graphic and visual interactions, were used with success. For example, Playground [1] uses a biological metaphor in which objects are artificial animals. Alternate Reality Kit [3] is an animated environment to create interactive simulations. It is based on a physical metaphor. Finally, the Rehearsal world [2], implemented in Smalltalk-80 and devoted to non-programmer curriculum designers, introduced a metaphor in which the unfamiliar concepts of programming have real-world referents: a world of a theatre where performers interact with one another on a stage by sending cues.

However, the programming activity of those systems is limited since the educational goal focusses mainly on animation and simulation by modifying parameters. The Rehearsal World allows the creation of new composite performers built from primitive ones and the definition of cues by writing simple code. More complex programming has to be written in Smalltalk directly. The PLAY system [4], which also uses the theatre metaphor, has added an iconic programming language simple enough to be accessible to preliterate children.

The context of this research is to help novices to learn Smalltalk programming. Smalltalk lacks powerful tools to ease definition and organization of complex objects. Consequently, we have decided to take again the theatre metaphor and to fully extend it to the object paradigm, in order to help learners to define and organize objects. Our system, called *Moliere*, is a visual object oriented programming environment built on top of Objectworks\Smalltalk. It deals with two different kinds of interactive objects, the logical objects which are decomposition objects (theatre, play, scenario, scene and cues) and the physical objects (performers, decor, costumes) which receive the cues. Every object has a graphical representation that is a box labelled with its name. The programming task consists in producing a theatre play. Several productions can be proposed for a play as several different algorithms or programs can solve a similar problem. A scenario, to which a script will be associated, is broken up into scenes which correspond to the various steps of the development of an application. A scene puts into play a set of characters or performers in a decor. Performers are active objects which know cues and which are auditioned. The scenes can be thought as reusable components for other programs. We see that a classical decomposition methodology is inherent, here, in the metaphor.

In addition to presentation windows which allow visual interaction with objects, we have designed a simple visual language to program the behaviours of objects (i.e. the methods). The visual syntax is based on labelled boxes. Our system does not follow the dataflow model, as do most visual languages, because this induces a functional or procedural

programming style. We wanted to keep explicitly the object-oriented paradigm. Our visual language is a close mapping of Smalltalk. Moreover, in Moliere the user interacts with instances directly. Indeed, Smalltalk is not a typed language and the increase of interaction given to objects by visual handling leads us to partially hide the class concept. New objects can be created by cloning and the method dictionary is accessible from an instance. Thus, the user, while exploring pre-existing classes, interacts with instances which have values for their instance variables and which can respond to messages.

We will show how to develop examples with Moliere and how the use of the metaphor allow the organization of the different steps of a program. This system is designed for high school students and educators. Although we could not carry out an experiment, we think it could be used as an effective programming environment in which to start object-oriented programming and to design interactive software.

### References

- [1] Fenton, J., Beck, K. (1989) Playground: an object oriented simulation system with agent rules for children of all ages, *OOPSLA '89 conference proceedings*, pp. 123-137.
- [2] Gould, L., Finzer, W. (1984). Programming by Rehearsal, *Byte*, 9 (6),187-210.
- [3] Smith, R.B. (1987) The Alternate Reality Kit- An animated Environment for Creating Interactive Simulations, *proceedings of the IEEE workshop on Visual Languages*, Dallas, June 1986,pp.99-106.
- [4] Tanimoto, S.L., Runyan, M.S. (1986) PLAY: an iconic programming system for children, in *Visual Languages* S.K.Chang & al. eds., Plenum Press, pp. 191-205.