

Visual Logic Programming Method Based on Structural Analysis and Design Technique

Alexei A. Morozov

Institute of Radio Engineering and Electronics RAS
Mokhovaya 11, Moscow, Russia, 125009
morozov@cplire.ru
<http://www.cplire.ru/Lab144/>

The convergence of logic and visual languages is recognized as a promising approach for both the logic programming automation and enhancement of visual modeling/programming methods. The distinction of our approach is in that we unite the Actor Prolog concurrent object-oriented logic language [1,2,3] with the Structural Analysis and Design Technique (SADT) diagrams to obtain new issues in the following areas:

1. Functional modeling of complex systems;
2. Rapid prototyping of artificial intelligence applications;
3. Graphic user interface management.

The SADT diagrams (also called the IDEF \emptyset diagrams) are a variety of functional diagrams and are widely applied for analysis and design of complex systems. Note that the main idea of our approach is in use of the logic language for analysis/animation of the SADT diagrams, but not in use of the visual notation for automation of programming. That is why we have employed SADT, but not the UML language that has much more specialized destination.

Our method of visual programming/modeling includes the following scheme of logic program design:

1. Standard SADT diagrams are used to develop a graphic description of the software system. SADT description is a hierarchy of blocks that receive and pass data flows (see an example of SADT diagram on Fig. 1).
2. Each elementary block of the SADT model is put into correspondence with a logic description in the form of a certain class of Actor Prolog. The source text in Actor Prolog can be written by a programmer or taken from the library of reusable modules (components).
3. The graphic description of the software system is automatically translated into the text in Actor Prolog. The syntactic means of Actor Prolog make possible to implement the block-hierarchical structure and links between blocks of the diagrams in the form of communicating processes.
4. Assembling the automatically created text and descriptions of elementary blocks, we obtain a ready-to-use program in Actor Prolog.

Experimentation with visual programming has shown that the diagrams can be conveniently used not only as a visual programming language but simply as a graphic user interface. At present, visual programming system automatically

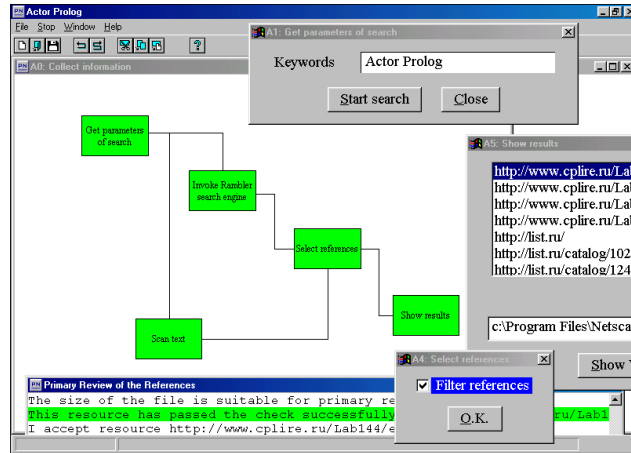


Fig. 1. An example of user interface based on SADT diagram.

creates visual interface of a logic program on the basis of source SADT diagrams (see example of user interface on Fig. 1).

The individual blocks of visual user interface based on the SADT diagrams are implemented by using parallel processes of Actor Prolog. As a rule, each elementary block of a diagram has its own dialog box that can be opened by the click of a mouse. The color of the block changes automatically depending on the state of the corresponding process. A user may interact with the blocks of the diagram in any order. Repeated alteration of any parameters entered earlier is also possible.

We applied the visual programming method for logic programming of Internet agents [4]. It was shown that the tools for visual programming based on SADT and the object-oriented logic approach substantially simplify and accelerate creation of complex logic programs.

This work was supported by RFBR, project no. 06-07-89302.

References

1. Morozov, A.: Actor Prolog: an object-oriented language with the classical declarative semantics. In Sagonas, K., Tarau, P., eds.: Proc. of the IDL'99 Int. Workshop, Paris, France (1999) 39–53 Available at: <http://www.cplire.ru/Lab144/paris.pdf>.
2. Morozov, A., Obukhov, Y.: An approach to logic programming of intelligent agents for searching and recognizing information on the Internet. Pattern Recognition and Image Analysis **11**(3) (2001) 570–582 Available at: <http://www.cplire.ru/Lab144/pria570m.pdf>.
3. Morozov, A.: Getting Started in Actor Prolog. IRE RAS (2002) Available at: <http://www.cplire.ru/Lab144/start/>.
4. Morozov, A.: Development and application of logical actors mathematical apparatus for logic programming of Web agents. In Palamidessi, C., ed.: Logic Programming: 19th International Conference, ICLP. LNCS 2916, Springer-Verlag (2003) 494–495