KNOWLEDGE BASED TOOL FOR MODELING DYNAMIC SYSTEMS IN EDUCATION

Krassen Stefanov, Galina Dimitrova
Faculty of Math. and Informatics
University of Sofia
James Bouchrier str. 5A
1126 Sofia, Bulgaria

Abstract.
Dynamic systems are often used in education: in physics, chemistry, biology, history, etc. This paper describes a programming tool for modelling dynamic systems which can assist the teaching and training process. The tool aims at supporting the creation of computer models which can be used for educational games and simulations. The tool is built using the logic programming language Prolog.

1. INTRODUCTION.

There are many examples of dynamic systems used in the school curriculum. Knowledge-based environments (KBE) are appropriate enough for modelling such systems. A lot of researchers have concentrated their efforts to construct suitable KBE and to investigate how their usage can help in studying dynamic systems. In spite of the fact that many good KBE exists, they cannot be used directly in Bulgarian schools because:

a) the hardware is very specific (PC-like computers causing wide range of hardware and software problems);

b) the usage of Cyrillic alphabet raises another kind of problems;

c) software support and further development of foreign software systems is very problematic, if possible at all, because of the poor quality of links and communications.

This leads us to the need of creation of our own KBE for use in education. The tool represented here is a part of such an environment [Dim Stef 91].
In this paper we represent a formal model for describing dynamic systems. We outline the structure of the tool (its components and relations with other tools) and propose how this tool can be used in the classroom.

2. INFORMAL MODEL FOR DESCRIBING DYNAMIC SYSTEMS.

We consider each dynamic system as represented by a state space: a set of states and actions.
Each state is described as a set of objects, their properties and relations.
Each action causes either change to another state (also called operator) or some local change in the state, concerning objects and properties. An action can be caused either by conditions presented in the state (relations concerning objects and their properties), or by external intervention (student, teacher or another agent).
Each object has his own graphical representation.
To each action a specific dialogue and graphic effect or animation can be attached.
Global parameters are objects with various types of values: interval, set, predicate, function.
Relation can be:
-property (predicate representing object's property);
-relationship between objects;
-condition (predicate representing the relationship between state and action - when the state fires a certain action);
-transition (predicate representing the relationship between action and state - when the action causes the appearance of a new state);
-constraint (legal or illegal values of parameters, properties, actions).

There are two types of relations: typed and arbitrary. Typed relations are described through facts concerning concrete objects involved in the relation. Arbitrary relations are defined by clauses given from the author.
The proposed tool uses the frame mechanism [Min 75] to describe formally a dynamic system as a state space. More detailed description of the frame system being used can be found in [Dim Stef 91].
Each state and each object is described by a frame including its graphic representation, a set of relations, a set of admissible actions and constraints. Each frame - action includes action's type, an animation sequence, a sequence of dialogues related to the action, changes of global parameters' values, etc.

3. FORMAL MODEL FOR DESCRIBING DYNAMIC SYSTEMS.

The system is represented as a set of frames. Below we give a formal description of each frame and its components.

Frame SYSTEM - has following slots:
-type (game, simulation, model);
-initial state;
-final state(s);
-target (rule determining the target: to achieve a final state or certain relation to hold; to carry out an experiment; etc.).

Frame STATE - has following slots:
-type (initial, final, forbidden or visited);
-graphic representation;
-set of objects available in the state;
-set of holding relations;
-set of actions and the conditions activating them;
-set of operators and corresponding transitions;
-set of constraints.

Frame OBJECT - has following slots:
-graphic representation of the object;
-set of properties;
-set of constraints;
-set of legal actions;
-set of visual effects.

Frame ACTION - has the following slots:
-type (operator, external action, movement);
-sequence of pictures representing action's effects;
-sequence of dialogues;
-modifications of global parameters' values;
-changes of objects' properties and relations;
-transitions corresponding to the action (in case of operator);

Frame RELATION - has the following slots:
-type (property, condition, constraint, relationship);
-predicate name (the predicate determining the relation);
-scope of relation's validity (in a given state only, in dependence fron a given property, permanently or temporary, etc);
-flag indicating whether the relation is on or off.

Frame GRAPHIC REPRESENTATION - has the following slots:
-background picture;
-set of objects (for each: its name and position);
-flags indicating the ways of appearance and disappearance;
-visual effects;

Frame DIALOGUE - has the following slots:
-type (user's initiative, system's initiative, mixed);
-set of questions and conditions;
-set of answers and conditions;

Frame VISUAL EFFECT - has the following slots:
-type (colour effect, animation, representation effect, etc.);
-duration;
-sound effects;

4. ARCHITECTURE OF THE TOOL.

The tool has to provide:
-suitable user interface - editor or another tool for knowledge and data acquisition and building model's description;
-suitable ways for learning with the model - making an attempt to achieve certain goal; making experiments concerning model's behaviour when parameter values vary;
-appropriate and flexible internal representation of the model;
-transparency of model's functioning.
Important to the computer model are the graphic representations of objects, animation of their actions and transitions.

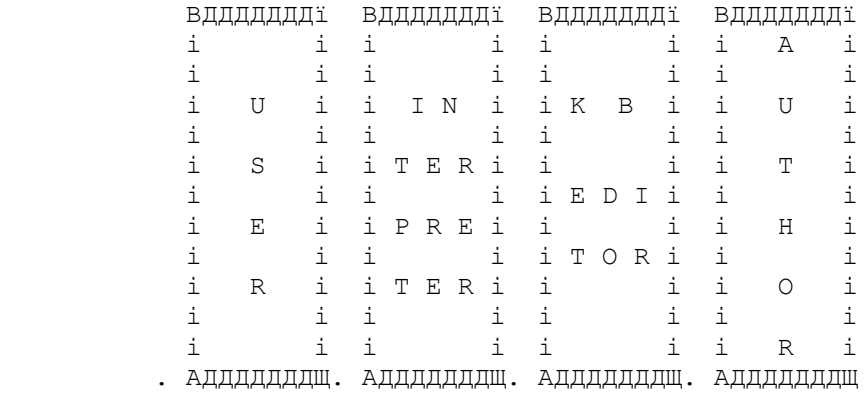

Fig. 1 shows the common architecture of the tool.

The implementation language is Prolog extended with predicates supporting object-oriented programming style and graphical facilities.
Briefly description of model's components follows. More detailed description of the underlying concepts can be found in [Dim Stef 91].

4.1. Knowledge Base.

The knowledge base (KB) is a collection of frames (classes and instances) and production rules, representing model's objects, their properties and relations between them, etc. The KB includes a set of global parameters and their legal values and a set of graphical representations.

4.2. Knowledge Base Editor.

The system is provided with a knowledge base editor which allows the user to build, modify and examine the knowledge base by using an interactive system of menus and dialog boxes.
The editor offers the following basic facilities:
-Initializing the process of creating a new knowledge base.
-Saving the knowledge base as a file containing Prolog clauses so that the internal knowledge representation can be examined and changed from within any text editor.

-Loading a previously saved knowledge base from a file.
-Manipulating the existing knowledge structure in the directions listed in section 3.3.
-Browsing the knowledge base: KB structure, frame structure, parameter values and etc.
-Searching through the knowledge base for objects with specified properties.
-testing the formal description of the model for completeness and consistency.
All these facilities are realized by the interpreter described bellow. The editor allows a user-friendly interface to the knowledge base.

4.3. Interpreter.
The interpreter manages and supports the KB.
The managing functions allow:
-to start the system;
-to save the current state;
-to restore a saved state;
-to interpret the model in various ways - beginner/advanced student, demonstration/training, experiment, etc.;
-to help the student.
The interpreter performs the basic knowledge processing operations :
-assertion of frames, instances and rules into the KB;
-assertion of properties to existing frame descriptions;
-generation of an instance from a given frame;
-exploration of the KB structure;
-modification of frames and instances;
-applying the rules to frames and instances.

4.4. Manipulating graphic representations.
The tool can use graphic representations created by standard graphical editors. A set of predicates of the interpreter is responsible to carry out the graphic interface. They allow:
-to load pictures (representations of objects, states) from files;
-to save a new picture in a file;
-to show a sequence of pictures;
-to change the pictures' background and foreground colour;
-to achieve some visual effects;
-to mix pictures with sound effects.

5. CONCLUDING REMARKS.

The tool for games and models can be used for creating educational computer games or models of objects, processes and phenomena. The tool uses a graphic extension to Prolog enabling various graphic commands to be used as built-in predicates. Its knowledge base contains key objects in the game or model, their properties, crucial phases, logic of shifting from one state to another, main aims and others.
The main directions in which the environment could be used are the following:
-Knowledge based systems and expert systems, with natural language interface, enabling students to work with different knowledge bases.
-Computer games and models for training in various areas and gaining practical skills and abilities.
-Creating intelligent tutoring systems in different areas.

References:

[Min 75] Minsky, M. (1975) A Framework for Representing Knowledge, in P. Winston (ed.), The Psychology of Computer Vision, New York.

[Dim Stef 91] Dimitrova G. and Stefanov K. (1991), Educational Knowledge Based Environment, in proceedings of the Sixth International PEG Conference: Knowledge Based Environments for teaching and Learning, Rapallo (Genova), Italy.