Hybrid Intelligent Systems

Lecture 9

ESWin – toolkit for development of hybrid expert systems

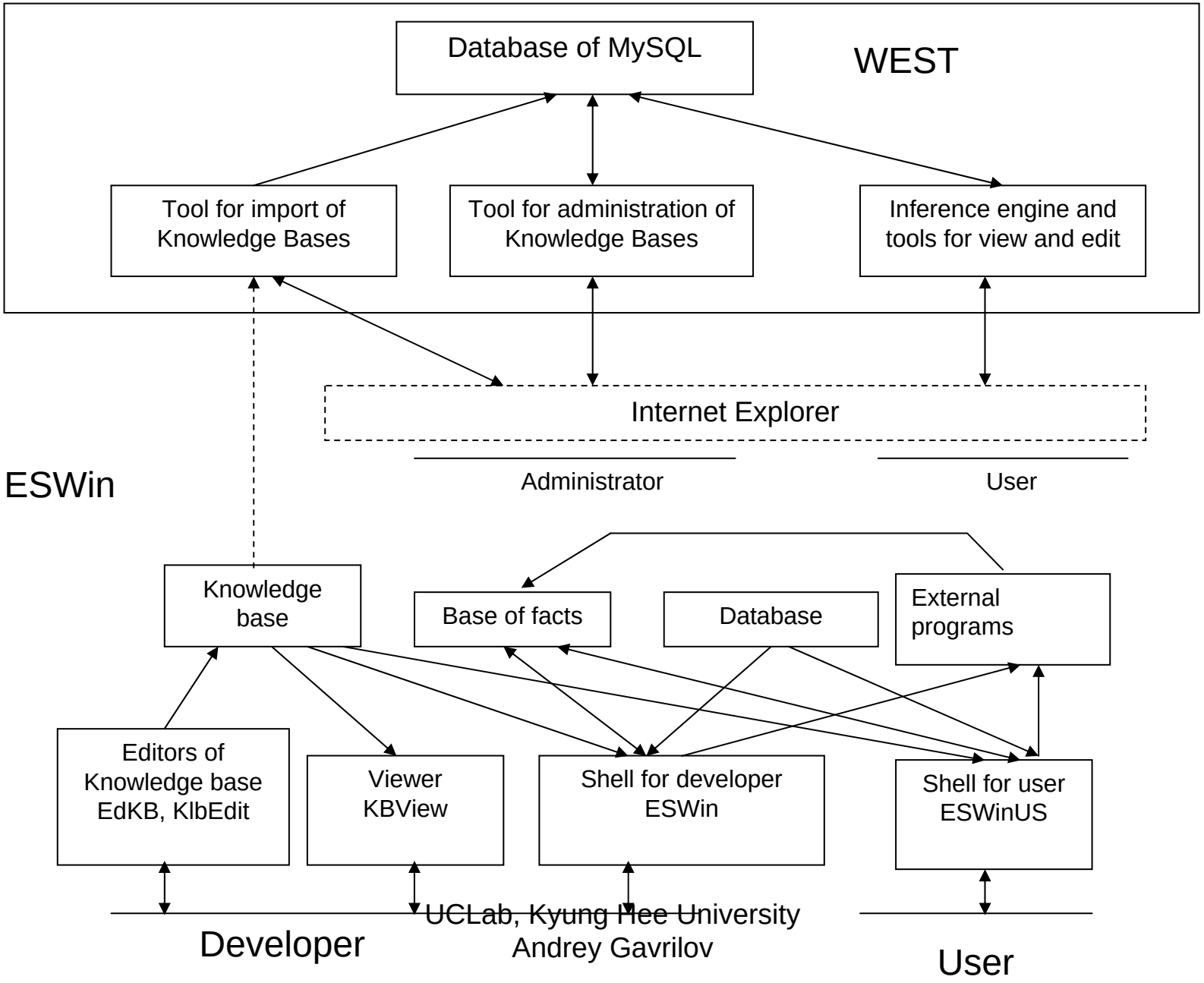
ESWin

- Consists of:
  - Language for description of knowledge base
  - Expert shell for developer
  - Expert shell for end user
  - Two kinds of editors of Knowledge Base (KB)
  - Program utility for view and diagnostics of KB
  - Program utility for improvement of structure of KB
- Solving of task by backward fuzzy inference
- Aims to using for development of expert systems for diagnostics, identification, support of making of decisions
- Used in several Universities of Russia for teaching of AI and related courses
- Downloaded (demo) by several foreign users for study
- Demo of ESWin is used in KHU in course “Technologies of Expert Systems”

Expert shell supports:

- Knowledge representation by
  - Rules
  - Frames,
  - Linguistic variables
- Solving of tasks by backward fuzzy inference
- Nonmonotonic reasoning
- Usage of facts from databases by SQL-query
- Usage of graphics for comments to questions or as results of rule’s execution
- Execution of external programs during inference

Structure of ESWin and its ability of connection with WEST



Expert shell for Internet WEST

- Component WEST 1.0 may be used as independent product and consists of:
  - Backward fuzzy inference engine,
  - Tool for administration of Knowledge Base,
  - Tool for import of Knowledge Base from ESWin.
- Developed in MySQL and PHP
- Accessible for demonstration from http://vt.cs.nstu.ru/~expsystem/
- That's all about WEST here

Sources of facts for inference in ESWin

- Dialog with user
- Databases, SQL-query forming automatically during dialog
- External special programs been developed in case that capabilities of ESWin are not enough for solving of task
- For example, as external program may be any neural network or simulation program

Knowledge base

- Consists of:
  - TITLE = ‹name of Expert System›
  - COMPANY = ‹name of organization - owner of ES›
  - Frame with name Goal with names of tasks solved by expert system
  - Other frames describing of domain
  - Rules for solving of tasks
  - Descriptions of linguistic variables (if ones are used in expert system) in separate file

Structure of frame

FRAME (‹ type of the frame ›) = ‹ a name of the frame ›
PARENT: ‹ a name of the frame - parent ›
OWNER: <a name of the frame - owner>
‹ A name of slot 1 › …..
….
‹ A name of slot i › (‹ type of slot ›) [‹ a question of slot ›?]
{‹ the comment of slot ›}: (‹ value 1 ›; ‹ value 2 ›; …; ‹ value m ›)
…
‹ A name of slot n › …..
ENDF

Types of frames:
- Class
- Instance
- Template

Types of slots:
- Symbol
- Number
- LV – linguistic variable

Examples of frames

Frame=Parameters
  Parent:
  Area: (Computer Science; Technology;
Medicine)
  Task: (CAD; CAM; Monitoring; Diagnostics)
EndF

Frame=Initial data
Parent:
Type of a body (symbol) [Choose type of a body]: (Sedan; Cabriolet;
Unified; Hatchback; Minivan)
Type of a box of transfers (symbol) [Choose type of a box of transfers]:
(Automatic; Manual)
Type of the engine (symbol) [Choose type of the engine]: (Diesel;
Petrol)
Price (number) [how many money you are ready to spend?]:
EndF

Structure of rules

RULE ‹number›
‹condition 1›
‹condition 2›
…
‹condition m›

DO
‹conclusion 1›
‹conclusion 2›
…
‹conclusion n›
ENDR

Relations in conditions can be:

EQ	or =	Equal;
GT	or >	It is more;
LT	or <	It is less;
NE	or <>	Not equal;
IN		Two frames are connected by the relation "part of" (there is a connection through slot OWNER).

Relations in the conclusions can be:

EQ	or =	Equal (creation of the fact - slot in a frame-instance);
IN		Including in the frame-owner (creation of connection - slot OWNER in the frame-instance);
DL		Remove of slot in a frame-instance;
EX		Execute of the external program;
FR		Output of a frame-instance;
GO		Execute of the rule;
MS		Output of the message to the screen;
GR		Output to the screen of a graphic file (formats *.gif, *.bmp, *.avi or *.htm).

Examples of rules

Rule 1
=(Initial data.Type of a body; Unified) 100
=(Initial data.Type of the engine; Diesel) 100
=(Initial data.Type of a box of transfers; Automatic) 100
<(Initial data.price; 1000) 100
Do
= (Goal.To buy the car; Under your choice approaches Toyota Caldina 1988) 100
EndR

Rule 1
EQ(Parameters.Area; Medicine)
EQ(Parameters.Task; Diagnostics)
Do
EQ(Knowledge representation method; Fuzzy Rules) 90
EQ(Knowledge representation method; Frames) 95
EQ(Tool for Developer; ESWin) 95
EndR

Linguistic variables

- Parameters describing of linguistic variable:
  - Name
  - Set of symbolic values
  - For every symbolic values
    - Minimal numeric value
    - Maximal numeric value
    - Number of values of membership function
    - Set of values of membership function

Example of usage of linguistic variable

```
Frame=You
  Parent:
  Employment: (Unemployment; Engineer; Businessman)
  Age (lingvar) [How are You old ?]: (young; old)
EndF

Frame=She
  Parent: Women
  Age (lingvar) [How is her age?]: (young; middle age)
EndF

        Rule 1
          EQ(You.Age; young)
          EQ(She.Age; young)
          EQ(You.Employment; Businessman)
        Do
          EQ(Your chance of success is; Good)
        100
        EndR
```

Example 1 of KB in ESWin (fragment)

```
TITLE=Demo Expert System for Creating of ES
COMPANY=CopyRight 2000 Insycom Ltd.

Frame=Goal
  Parent:
  Knowledge representation method: ()
  Tool for Developer: ()
EndF

Frame=Parameters
  Parent:
  Area: (Computer Science; Technology; Medicine)
  Task: (CAD; CAM; Monitoring; Diagnostics)
EndF

Rule 1
  EQ(Parameters.Area; Medicine)
  EQ(Parameters.Task; Diagnostics)
Do
  EQ(Knowledge representation method; Rules with Fuzzy) 90
  EQ(Knowledge representation method; Frames) 95
  EQ(Tool for Developer; ESWin) 95
EndR

Rule 2
  EQ(Parameters.Area; Computer Science)
  EQ(Parameters.Task; Monitoring)
Do
  EQ(Knowledge representation method; Rules) 100
  EQ(Tool for Developer; C++) 100
EndR

Rule 3
  EQ(Parameters.Area; Technology)
  EQ(Parameters.Task; Monitoring)
Do
  EQ(Knowledge representation method; Rules with Fuzzy) 80
  EQ(Tool for Developer; ESWin) 80
  EQ(Tool for Developer; C++) 70
EndR

Rule 4
  EQ(Parameters.Area; Technology)
  EQ(Parameters.Task; CAD)
Do
  EQ(Knowledge representation method; Frames) 100
  EQ(Tool for Developer; Lisp) 95
  EQ(Tool for Developer; ESWin) 70
EndR

Rule 5
  EQ(Parameters.Area; Technology)
  EQ(Parameters.Task; CAM)
Do
  EQ(Tool for Developer; ESWin) 90
EndR
```

Example 2 of KB in Eswin (fragment)

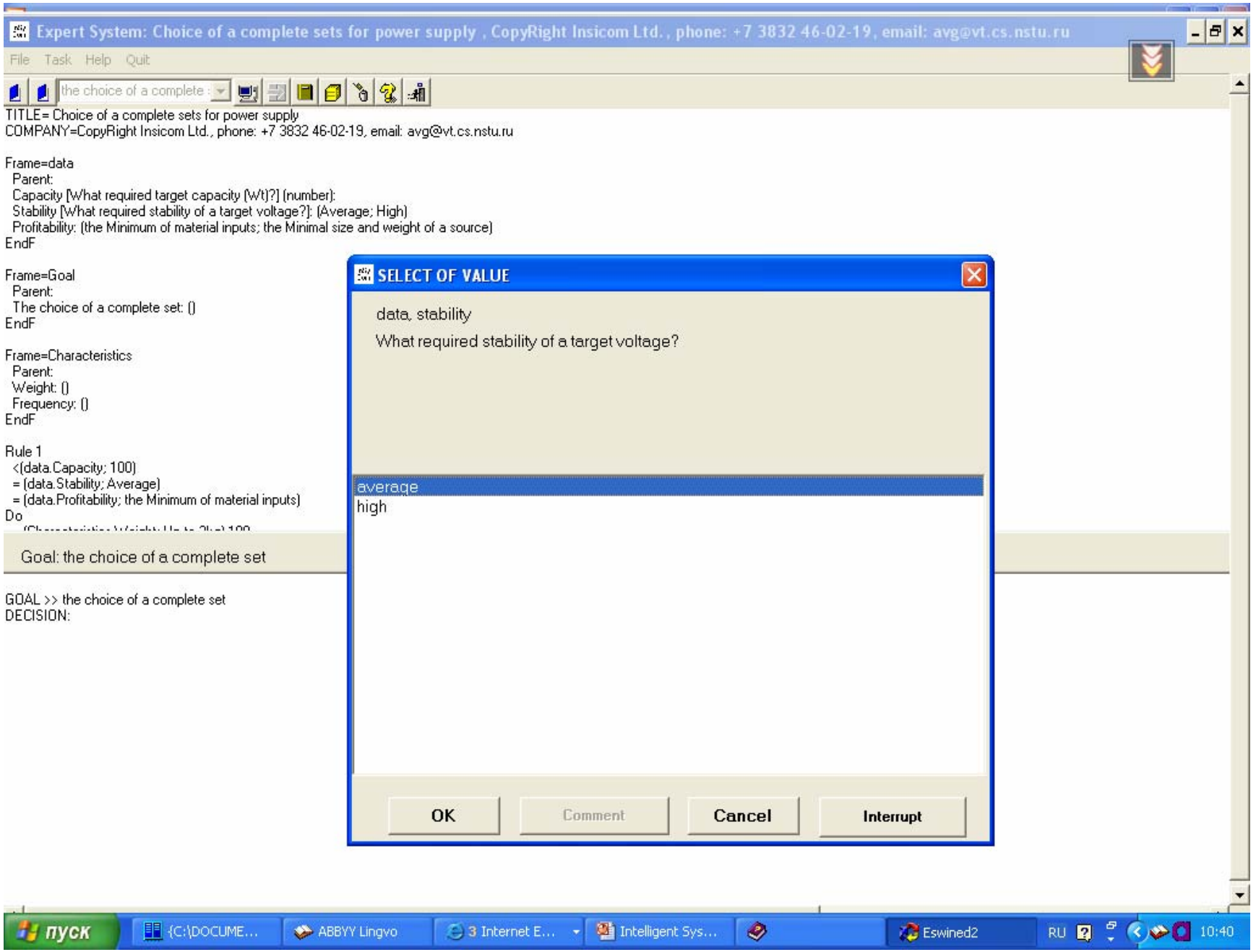
```
TITLE=Test of use "any"
COMPANY=Insycom Ltd.

Frame=Goal
  Parent:
  Test: ()
EndF

Frame=Features
  Parent:
  Area: (Office; CAD; Games)
EndF

Frame=Components
  Parent:
  Processor: (above 3 GHz; to 3 GHz; to 1.5 GHz)
  Monitor: (17"; 21")
EndF

Rule 1
  EQ(Features.Area; Games)
Do
  EQ(Components.Processor; above 3 GHz) 100
  EQ(Components.Monitor; 17") 100
EndR

Rule 2
  EQ(Features.Area; Office)
Do
  EQ(Components.Processor; to 1.5 GHz) 100
EndR

Rule 3
  EQ(Features.Area; CAD)
Do
  EQ(Components.Monitor; 21") 100
EndR

Rule 4
  EQ(Components.Processor; any)
  EQ(Components.Monitor; any)
Do
  EQ(Test; Components are selected) 100
  FR(Action; Components) 100
EndR
```

Example 3 of KB in Eswin (fragment)

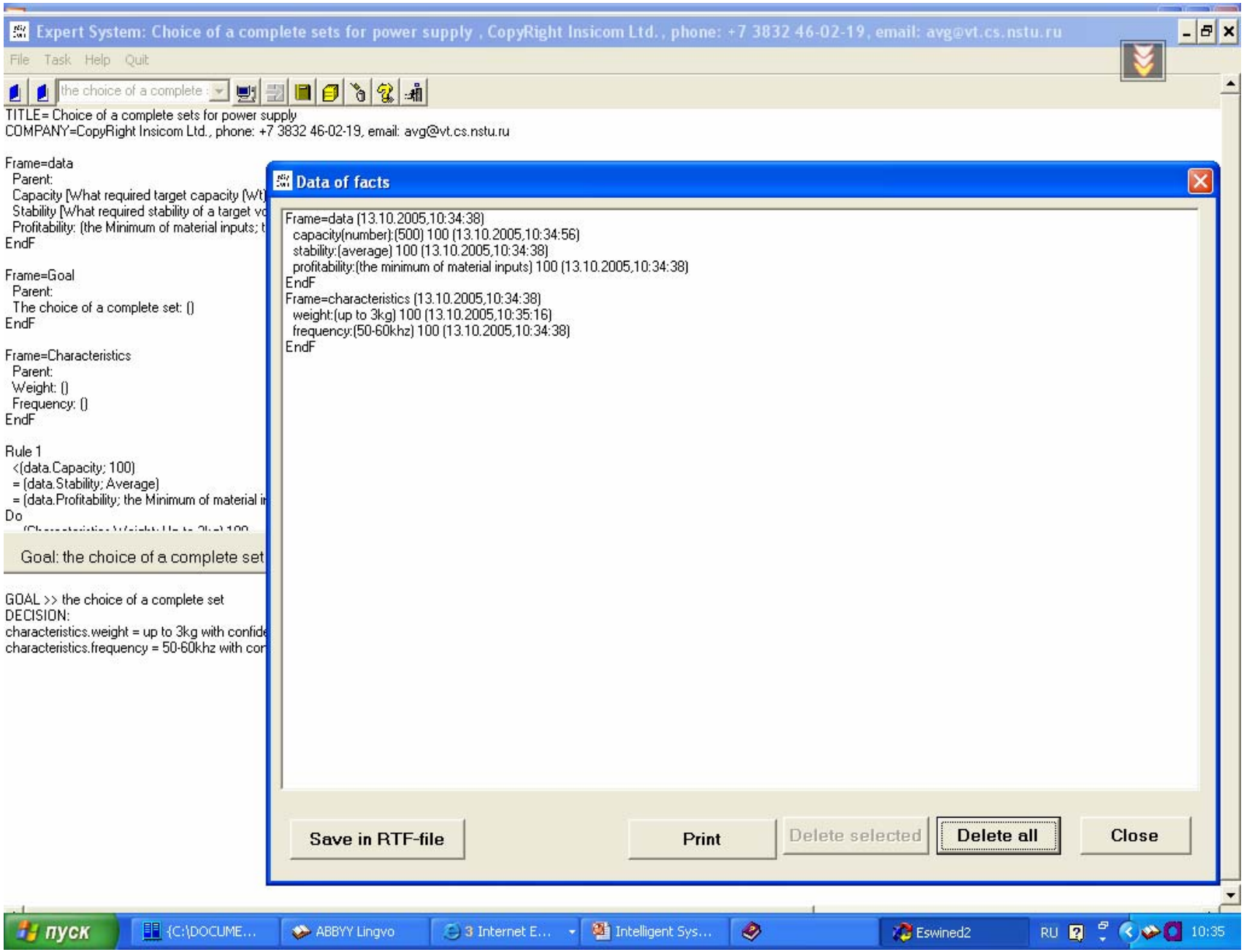
```
TITLE=Example of application with any  features

COMPANY= "Insycom Ltd."

Frame=Goal
   Charge of Salary:
EndF

Frame=Profit
   Fund for Salary [How much money may be directed
for salary?](number):
EndF

Frame=Men
    Position [Position of employee?] :
        *Name [Name of employee?]:
EndF

Frame=Name
   parent: Men
   *How many [How many to charge ($)?] (number) :
EndF
```

```
Frame=Salary
      SumSalary (number): (0)
EndF

Rule 1
            <(Profit.Fund for Salary; 500)
Do
            =(Charge of Salary; There is no sense)
EndR

Rule 2
            >(Profit.Fund for Salary; 499)
            =(Men.Name; any)
Do
      =(Charge of Salary; is successfull)
      =(Salary.SumSalary;
#Salary.SumSalary+Name.How much)
      FR(Frame; Salary)
EndR
```

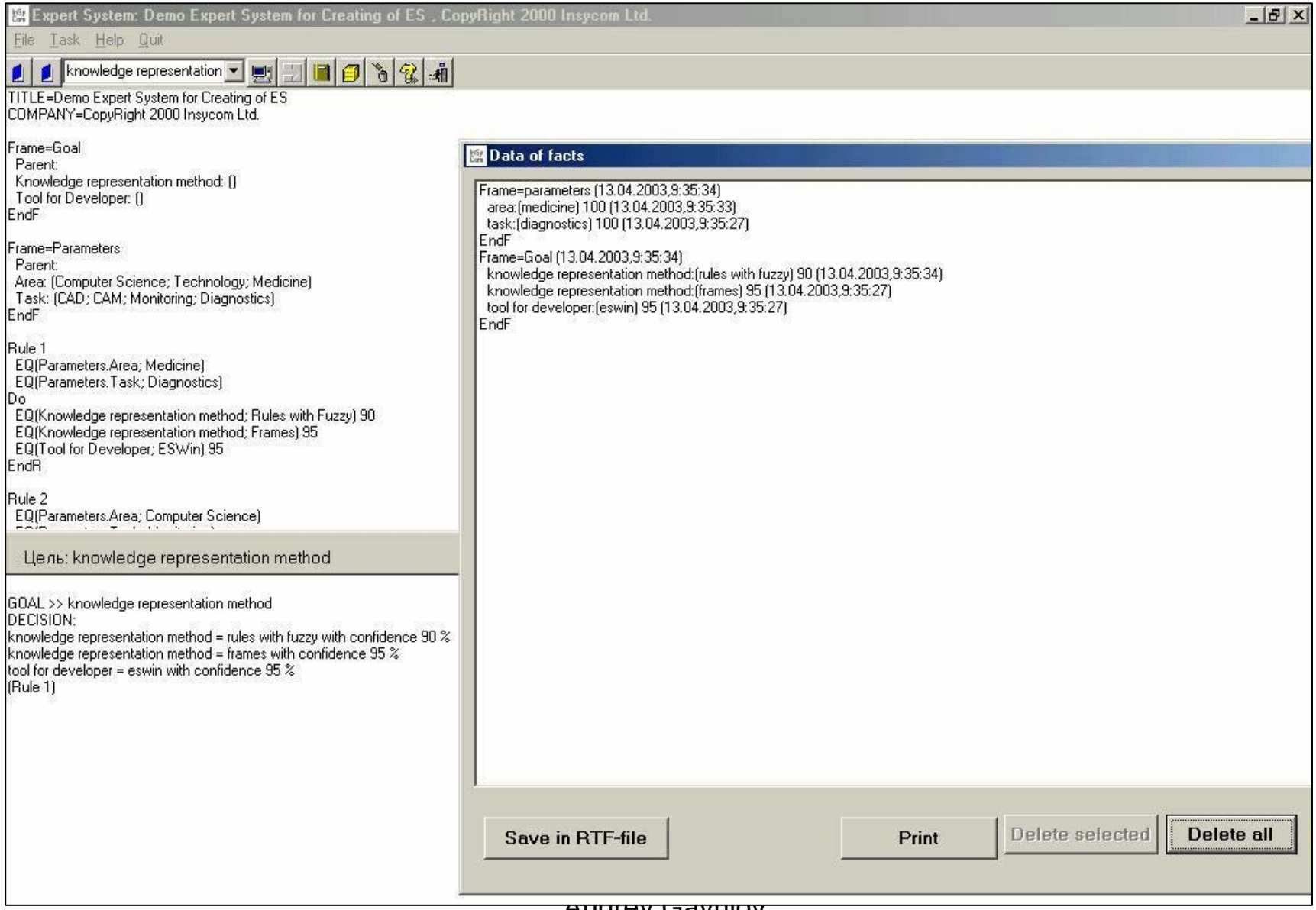
Expert shell ESWin for developers



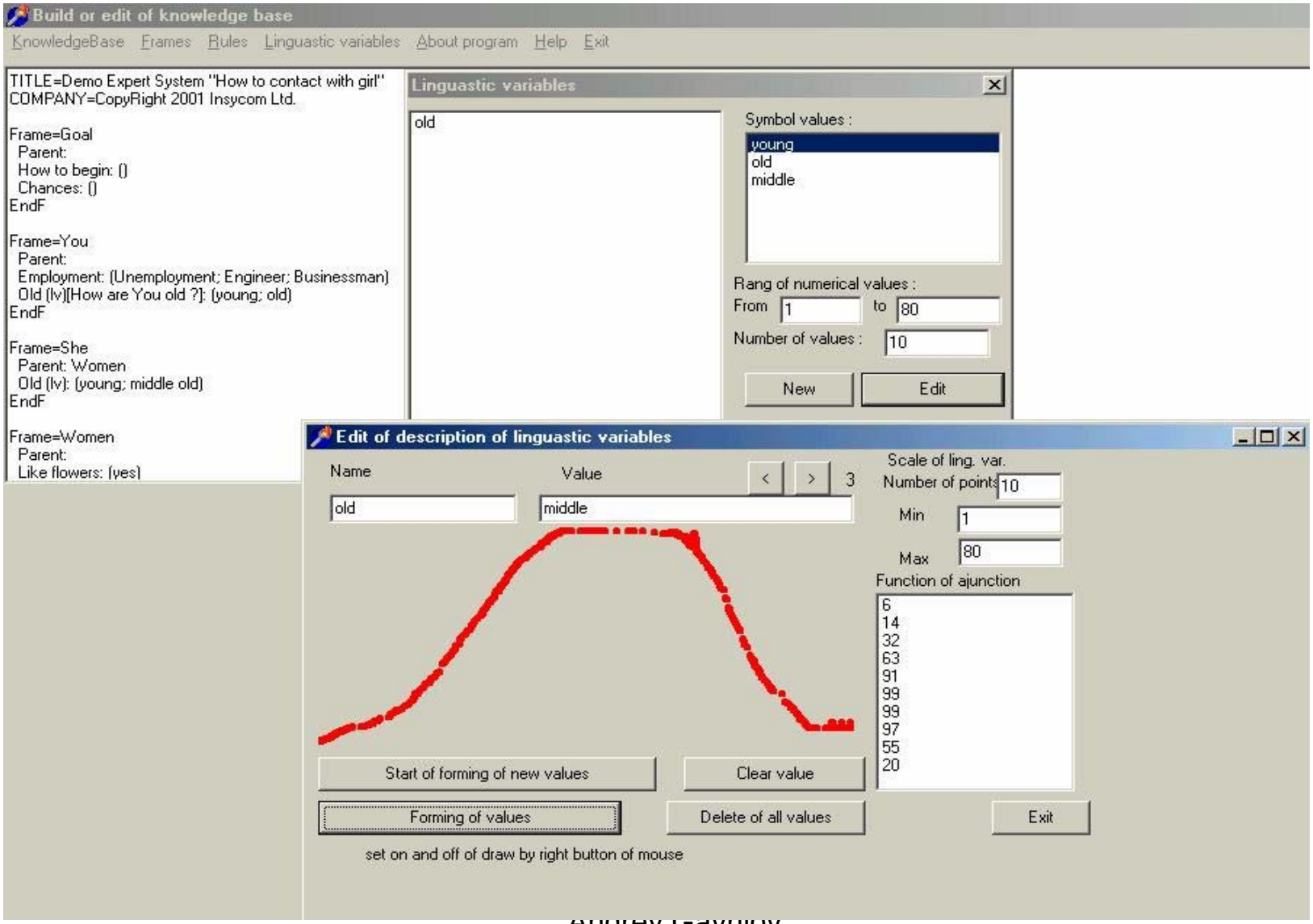
Expert shell ESWin for developers (2)



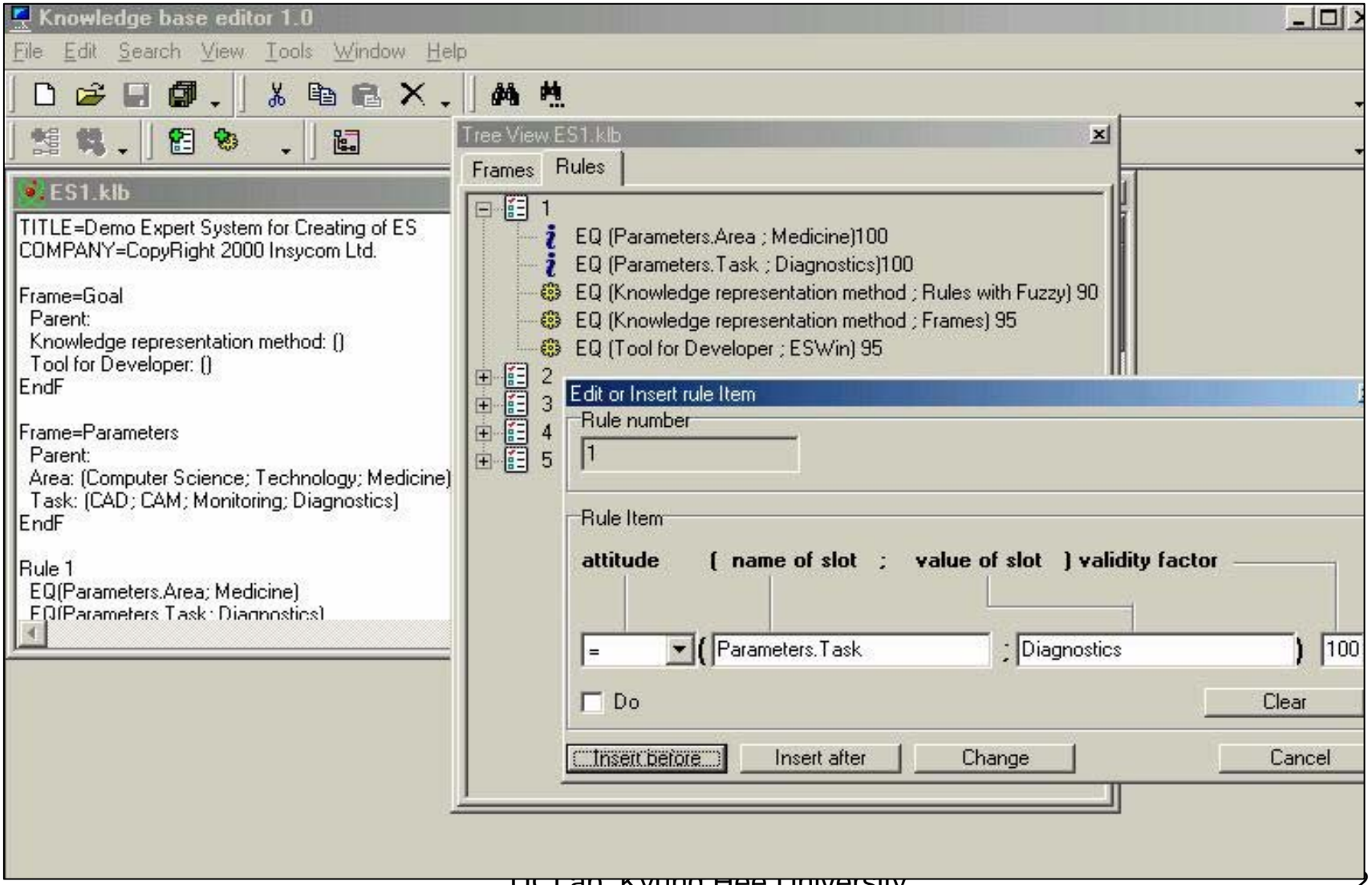
Editor of knowledge base



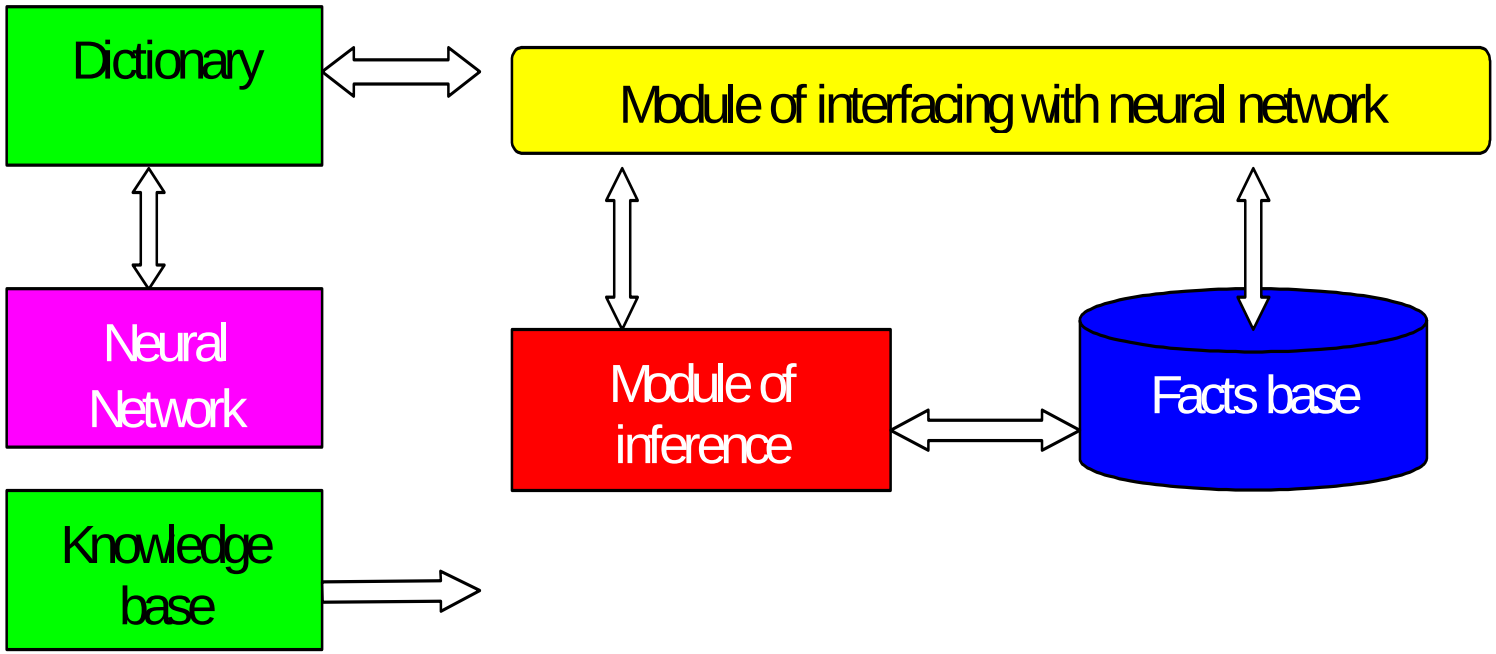
Editor of KB (editing of linguistic variables)



Other editor of KB



In present time the version of ESWin is developed for creating hybrid expert systems with neural networks

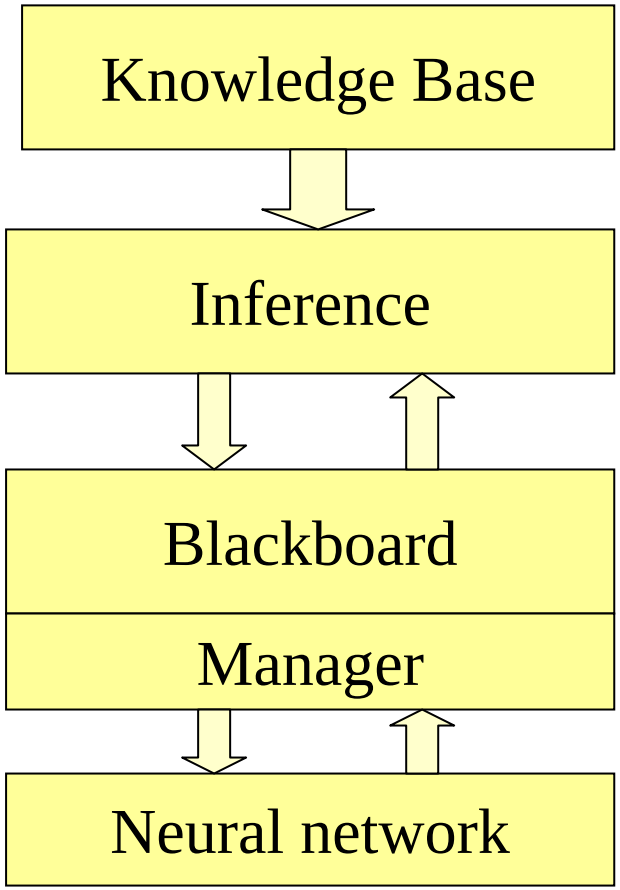


Example of description of protocol between Expert Shell and Neural Network in XML

```
<MODULE source=ES time=GetLocalTime()>
 <REQUEST target=NN source=ES dataType="frame">
 <DATA>

   <METHOD type="AskFactt">
    < FRAME name="Distance">
     <SLOT name="toObject" type="Number"></SLOT>
    </FRAME>
   < /METHOD >

</DATA>
</REQUEST>
</MODULE>
```

An architecture of “two-hemisphere” expert system (Gavrilov A.V., 1989)

- Level of store of knowledge
- Level of processing of data and knowledge
- Level of store of data
- Level of processing of signals and events



Possible functions of Neural Network

- Preprocessing of signals and data, received from external hardware, classification and clustering,
- Forming linguistic variables from examples,
- Generation of hypothesis based on facts from blackboard,
- Forming of associative links between facts for fast solving of task without inference

Future developments

- Development of collection of neural networks as modules of ESWin
- Introducing into inference the processing of temporal relations and entities “event”, “time”, “duration” to build of Real Time Expert Systems
- Implementation of inference engine as independent component for including in Real Time Systems
- Testing of proposed architecture on real tasks:
  - System for diagnostic and sorting of genuine leather
  - Control system of mobile robot (program model and real robot)

Opportunity of usage of ESWin in CAMUS

- Development of inference engine compatible with ESWin
- This one will allows to use ESWin for suitable building and debugging of knowledge bases for scenarios

Publications:

- Gavrilov A.V., Novickaja J.V. The Toolkit for development of Hybrid Expert Systems. - 5-th Int. Symp. “KORUS-2001”, Tomsk: TPU, 2001. - Proceedings. - Vol.1. - P. 73-75.
- Gavrilov A.V., Novitskaya J.V. The Architecture of the Hybrid Expert System. - / The 6-th Russian-Korean International Symposium on Science and Technology. KORUS-2002, Materials. - Novosibirsk, 2002. - Vol. 3.- P.70.
- Gavrilov A.V. Hybrid Intelligent Systems. – Novosibirsk, NSTU, 2003. – 162 p. (in Russian)
- Gavrilov A.V., Chistyakov N.A. An architecture of the toolkit for development of hybrid expert systems. // Proc. Of the Second IASTED Int. Multi-Conference ACIT-2005, Automation, Control and Applications, Novosibirsk, 2005. – Pp. 116-120.

Thanks