# Fuzzy Graphic Modeling library (MATLAB/Simulink library)

Viktor Pavliska
IRAFM, University of Ostrava, Czech Republic
`http://irafm.osu.cz`

December 19, 2011

## 1 Introduction

These days, a number of software developers create their software by drawing it. These engineers do not merely hand-translate a paper flowchart into text source code, or wield software tools to transform a UML diagram into C++. With graphical programming, the diagram is the source code, depicted as an arrangement of nodes connected by wires. Each piece of data flows through the wires, to be consumed by nodes that transform the data mathematically or perform some action such as I/O.

FGML (*Fuzzy Graphic Modeling Library*) is a MATLAB/Simulink[1] library which brings fuzzy modeling entities to the MATLAB environment in a graphical modeling manner. Fuzzy modeling tools may be applied to solve several kinds of tasks which have a common denominator vagueness. The following task belong among the typical ones solved by advanced fuzzy modeling tools:

- Fuzzy control
- Decision making
- Object classification and pattern recognition
- Approximation of functions

Practical problems are usually too complex and difficult to be modeled only by one technique, therefore a combination of several approaches is used to design system for solving the task. It turns out to be practically necessary to use such tools, which enable to design hierarchical and heterogeneous systems.

### 1.1 Visual programming

*Visual programming* [1] is programming in which more than one dimension is used to convey semantics. Examples of such additional dimensions are the use of multi-dimensional objects, the

---

[1] Simulink is a tool for simulating dynamic systems with a graphical interface specially developed for this purpose. Within the MATLAB environment, Simulink is a MATLAB toolbox that differs from the other toolboxes, both in this special interface and in the special "programming technique" associated with it. There is a further difference, in that the source code of the Simulink system is not open, but this is of no concern for our purposes. The goal of this chapter is to introduce simple manipulations with Simulink and to clarify the interaction of Simulink with MATLAB.

use of spatial relationships, or the use of the time dimension to specify “before-after” semantic relationships. Each potentially-significant multi-dimensional object or relationship is a token (just as in traditional textual programming languages each word is a token) and the collection of one or more such tokens is a *visual expression.* Examples of visual expressions used in visual programming include diagrams, free-hand sketches, icons, or demonstrations of actions performed by graphical objects. When a programming languages (semantically-significant) syntax includes visual expressions, the programming language is a *visual programming language* (VPL).

Although traditional textual programming languages often incorporate two-dimensional syntax devices in a limited way – an x-dimension to convey a legal linear string in the language, and a y-dimension allowing optional line spacing as a documentation device or for limited semantics (such as “continued from previous line”) – only one of these dimensions conveys semantics, and the second dimension has been limited to a teletype notion of spatial relationships so as to be expressible in a one-dimensional string grammar. Thus, multidimensionality is the essential difference between VPLs and strictly textual languages.

## 2 Installation

1. Extract archive file `fgml.zip` to arbitrary folder. This file contains library itself, license and one complex example of the library usage – hierarchic model ’houses’ which is mentioned later in this document.
2. Run MATLAB
3. Add instalation directory to MATLAB search path.

## 3 Library usage

Library `fgm_library.mdl` makes accessible algorithms from the software package LFLC 2000 [2] to the MATLAB environment.

It contains following blocks:

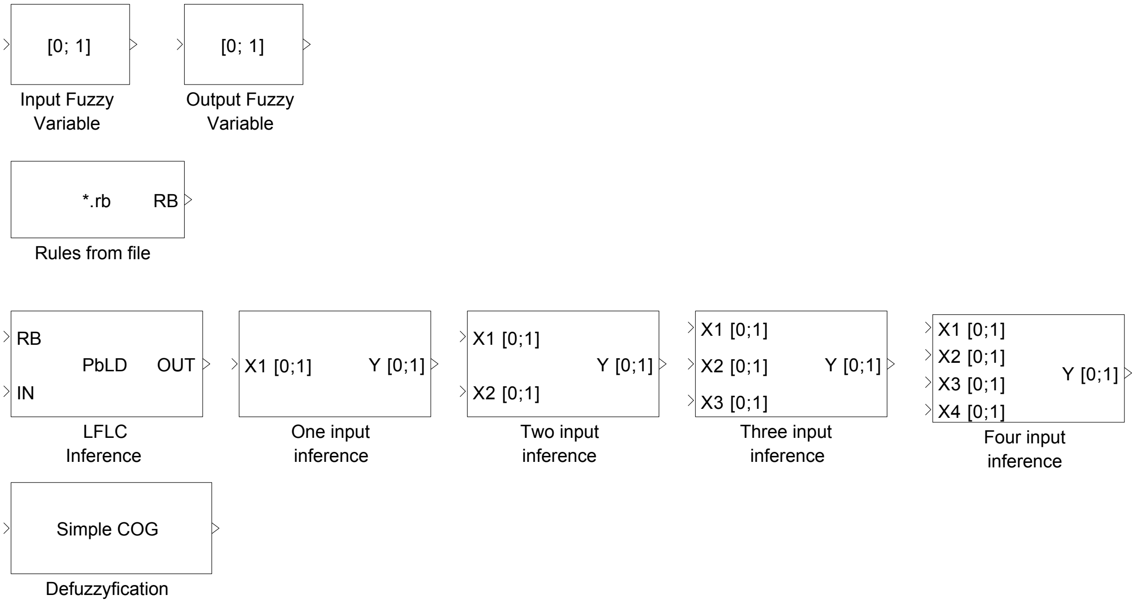
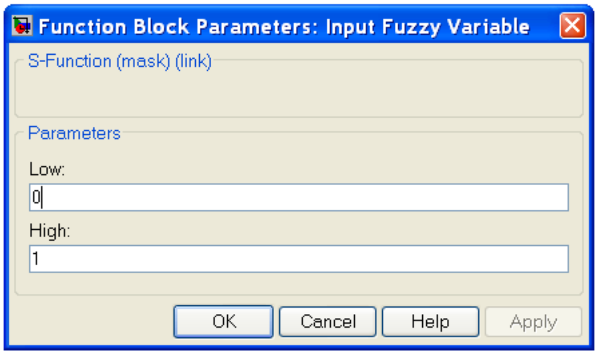


Image: FGML library content.

The user may drag-drop these blocks to his own models as he is used to do with standard Simulink library browser.

## 4 Description of library blocks

- **Input (Output) Fuzzy Variable:** Enable to define context of input (output) linguistic variables [4]. After double-clicking on the block user may specify low and high values for corresponding fuzzy variable in the following dialog box:



Low:
0

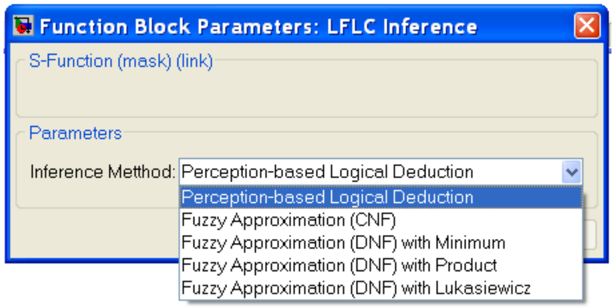
High:
1

Image: Dialog for definition of variable context.

- **Rules from file:** Allows the user to enter source file containing rules defining the behavior of linguistic description.

- **LFLC Inference:** Represents the core of computation of the output. It defines the main behaviour of the model. User may choose one from the following possibilities:

  1. Perception-based Logical Deduction (see [5, 6])
  2. Fuzzy Approximation (CNF)
  3. Fuzzy Approximation (DNF) with Minimum
  4. Fuzzy Approximation (DNF) with Product
  5. Fuzzy Approximation (DNF) with Łukasiewitcz


Function Block Parameters: LFLC Inference
S-Function (mask) (link)
Parameters
Perception-based Logical Deduction
Fuzzy Approximation (CNF)
Fuzzy Approximation (DNF) with Minimum
Fuzzy Approximation (DNF) with Product
Fuzzy Approximation (DNF) with Lukasiewicz

Inference Metthod: Perception-based Logical Deduction

Image: Selection of the inference method.

One up to Four input inference blocks are pre-made blocks which are shortcuts to predefined subblocks. For example Two input inference is defined for two variable input inference and looks like this:

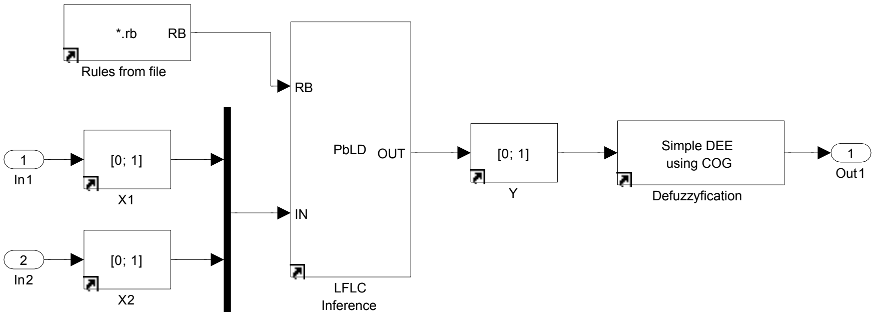
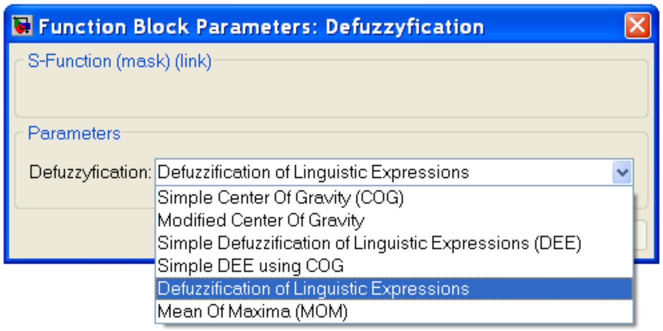


Image: Predefined two variable input inference.

- **Defuzzification:** The user may choose one of from the following defuzzification methods:
  1. Simple Center of Gravity (COG)
  2. Modified Center of Gravity (MCOG)
  3. Simple Defuzzification of Linguistic Expressions (DEE)
  4. Simple DEE using COG
  5. Defuzzification of Linguistic Expressions
  6. Mean of Maxima (MOM)


Function Block Parameters: Defuzzyfication
S-Function (mask) (link)
Parameters
Simple Center Of Gravity (COG)
Modified Center Of Gravity
Simple Defuzzification of Linguistic Expressions (DEE)
Simple DEE using COG
Defuzzification of Linguistic Expressions
Mean Of Maxima (MOM)

Defuzzyfication: Defuzzification of Linguistic Expressions

Image: Dialog for selection of the defuzzification method.

## 5 Example

For illustration purposes there is in the installation archive attached example called `houses.mdl`. This example contains several linguistic models connected to hierarchic knowledge base which may be used as demonstration of FGML capabilities.

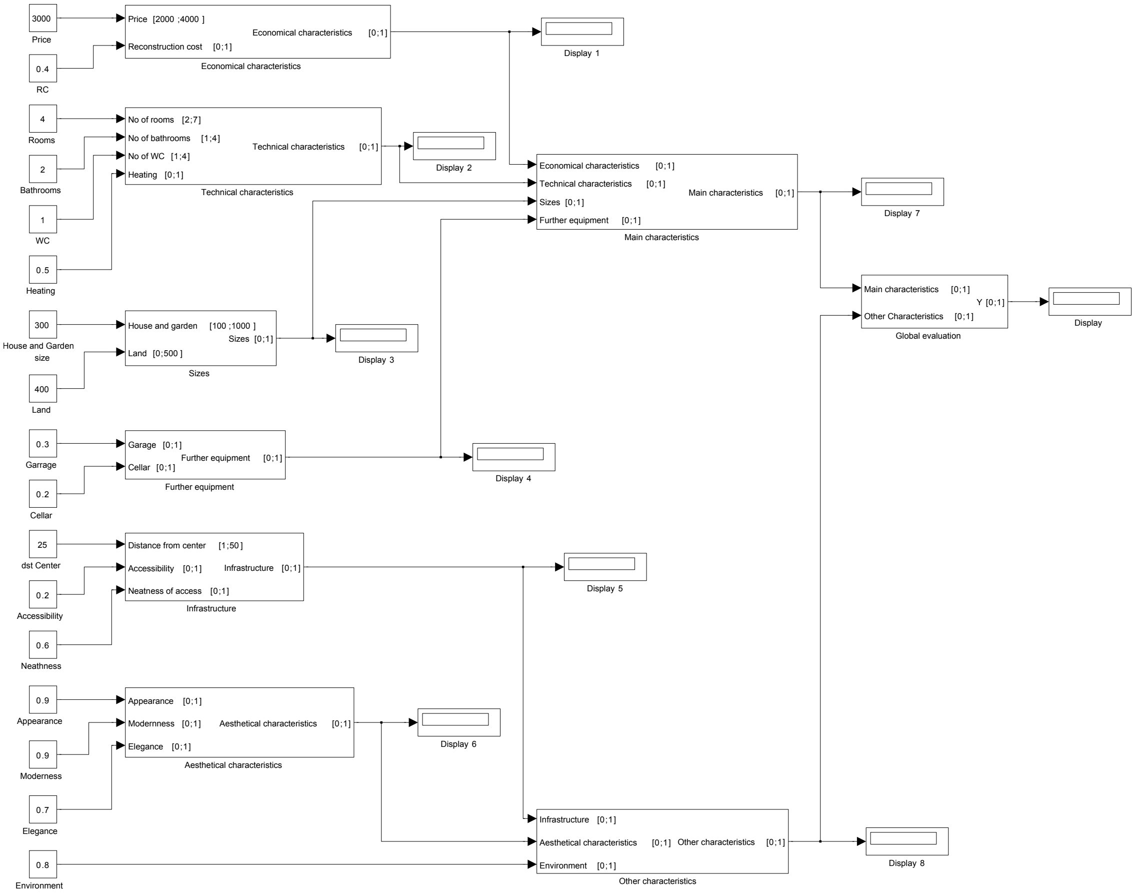


Image: Example of hierarchic model for the best house selection.

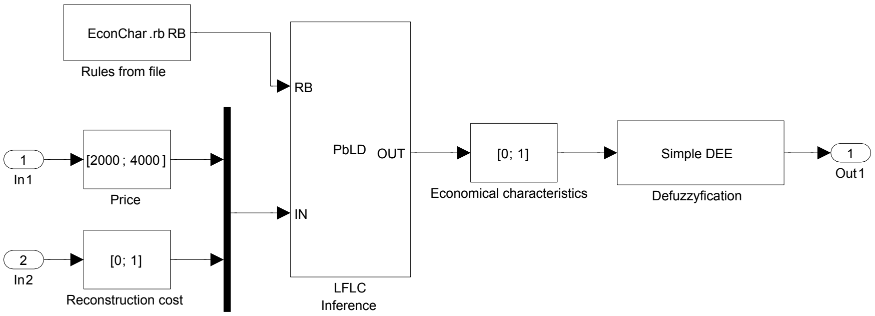


Image: Part of the above model grouping economy related characteristics.

# 6 Conclusion

A possibility to visually design own system or visually modify a predefined typical system composed from basic elements provided with the whole software package is the main advantage for users. Due to principles of graphical programming, users of such a system are provided by incomparable possibilities and by incomparable flexibility in comparison with classical systems.

# Acknowledgement

This software has been developed due to a financial support of the Research centre MŠMT 1M0572 DAR Data, Algorithms, Decision-making of the Ministry of Education, Youth and Sports of the Czech Republic.

# References

[1] Margaret M. Burnett: Visual Programming, *In: Encyclopedia of Electrical and Electronics Engineering (John G. Webster, ed.)*, John Wiley & Sons Inc., New York.

[2] Dvořák, A., Habiballa, H., Novák, V., Pavliska, V. (2003), The concept of LFLC 2000 – its specificity, realization and power of applications. *Computers in Industry*, sv. **51**, 269–280.

[3] Novák, V., Lehmke, S.(2006), Logical structure of fuzzy IF–THEN rules. *Fuzzy Sets and Systems*, **157**, 2003-2029.

[4] Zadeh, L. A. (1975), The concept of a linguistic variable and its application to approximate reasoning I, II, III, Inf. Sci. 199–257, 301–357, 43–80.

[5] Novák, V., (2005), Perception-Based Logical Deduction. In: Reusch, B. (Ed.): *Computational Intelligence, Theory and Applications*, Springer, Heidelberg, 241–253.

[6] Novák, V., Perfilieva, I. (2004), On the Semantics of Perception-Based Fuzzy Logic Deduction, *Int. Journal of Intelligent Systems*, **19**, 1007–1031.