UNIVERSITY OF OSTRAVA

Institute for Research and Applications of Fuzzy Modeling

Object Oriented Implementation of Fuzzy Logic Systems

Radim Bělohlávek, Antonín Dvořák, David Jedelský and Vilém Novák

Research report No. 11

August 31, 1998

Submitted/to appear:

Abridged version appeared in Camarinha-Matos, L.M. et al. (Eds.): Intelligent Systems for Manufacturing. Multi-Agent Systems and Virtual Organizations. Kluwer, Dordrecht, 1998, 589-594.

Supported by:

201/96/0985 GA AV ČR

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
Bráfova 7, 701 03 Ostrava 1, Czech Republic

tel.: +420-69-622 2808 fax: +420-69-22 28 28
e-mail: antonin.dvorak@osu.cz

Object Oriented Implementation of Fuzzy Logic Systems[1]

Radim Bělohlávek, Antonín Dvořák, David Jedelský and Vilém Novák
University of Ostrava,
Institute for Research and Applications of Fuzzy Modeling,
Bráfova 7, 701 03 Ostrava 1, Czech Republic.
e-mail: {belohlav, dvoraka, jedelsky, novakv}@osu.cz

Abstract

Various aspects of implementation of fuzzy software systems are discussed. Some theoretical considerations on basic notions are introduced and also our approach to implemetation is presented.

Keywords: fuzzy control, object-oriented programming

1 Introduction

This contribution has two main aims: to present main concepts which lead us to the creation of the software system LFLC (Linguistic Fuzzy Logic Controller) for fuzzy control, decision making and modeling, and to discuss object-oriented implementation of these concepts. We believe that our experience can be useful for those who work in the field of the soft computing software and, also to advanced users who want to understand better the theoretical background of the fuzzy control and modeling. We use C++ for programming and will use the common terminology used by the C++ programmers. The following discussion will concern only implementation of the kernel of fuzzy software system, i.e. basic classes and their data members and methods. We will not discuss the design of user interface. This paper is organized as follows: In Section 2, we present some theoretical concepts important for fuzzy control and modeling, Section 3 contains discussion about implementation of concepts from Section 2, and Section 4 puts forth some conclusions.

2 Linguistically oriented fuzzy control and modeling

Fuzzy sets are intended for modeling of situations where the phenomenon of vagueness is present. A typical case is modeling of the semantics of natural language descriptions (e.g. descriptions of complex systems). Recall that a fuzzy set in the universe set U is a mapping $A : U \mapsto [0, 1]$. For $x \in U$, the value $A(x)$ is interpreted as the truth degree of "x is a member of A". There are two basic kinds of representation of a fuzzy set—by the membership function or by the family of its α-cuts [3, 4]. Similarly as in the classical case, various operations may be performed with fuzzy sets [3, 4], based usually on the operations on the set of truth values (t-norms, t-conorms and derived operations). The most successful applications of fuzzy sets are in fuzzy control.

The role of fuzzy controllers is nowadays well acknowledged. They became popular for their transparency and for the fact that they offer solutions even in situations where classical controllers fail to succeed. Their idea has been proposed by L. A. Zadeh [12] and it has been applied to control by E. H. Mamdani. Recall basic facts: Even in situations where no exact mathematical description of the controlled process exists, experienced operators are still able to preform successfully the control. The operators are able to express the control strategy in the form of a linguistic description. Having a suitable formal apparatus for modeling of this linguistic description at disposal we are able to mimic the decision processes made by humans. The apparatus is provided by fuzzy set theory and theory of approximate reasoning. This is the background of all of the various types of fuzzy controllers. However, the linguistic point of view gets often lost. As a result, the declared transparency of the model decreases in such case. Especially non-transparent are techniques of adaptation of the controller where the fuzzy sets representing the linguistic expressions are arbitrarily modified. This is useful from the control-engineering point of view, however, the resulting model can be hardly interpreted on the level of linguistic descriptions.

[1]This work has been supported by the grant 201/96/0985 of the GA ČR.

The linguistic point of view is consistently pursued in the so called *linguistically oriented fuzzy control* (LOFLC) [7]. The main idea of LOFLC is that of keeping in mind two levels—the level of linguistic descriptions (syntax) and the level of the meaning of the descriptions (semantics). These two levels are connected as described below. However, we insist on the fact that at every time moment, the level of syntax should have the corresponding reflection in the level of semantics and, conversely (which is crucial), the level of semantics should have the corresponding reflection in the level of syntax. The formal agenda goes as follows. We suppose that the expert knowledge of the control is expressed by the set

$$\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$$

of linguistic control rules $\mathcal{R}_i$, for example

$$\mathsf{IF}\ \text{temperature}\ \mathsf{is}\ \textit{high}\ \mathsf{THEN}\ \text{turn the cap to the}\ \textit{right}\ .$$

The set of rules is called the *linguistic description* of the control. According to the information involved and the kind of the control action, P, PI, PD and PID control is usually distinguished. For example, in the case of PI control each $\mathcal{R}_i$ has the form[2]

$$\mathcal{R}_i := \mathsf{IF}\ e\ \mathsf{is}\ \mathcal{A}_i^e\ \mathsf{AND}\ \Delta e\ \mathsf{is}\ \mathcal{A}_i^{\Delta e}\ \mathsf{THEN}\ \Delta u\ \mathsf{is}\ \mathcal{B}_i^{\Delta u}. \tag{1}$$

Here, $e = y - v$ (y is the process output and v is the setpoint) and Δe denote the error and change of error (with respect to the previous time step), respectively, and the symbols $\mathcal{A}_i^e$, $\mathcal{A}_i^{\Delta e}$, $\mathcal{B}_i^{\Delta u}$ denote appropriate linguistic expressions. The variables e, Δe, and y are the so called fuzzy variables [4, 12]. The general form[3] of the considered linguistic expressions is

$$\mathcal{A} := [\langle\text{linguistic modifier}\rangle]\langle\text{atomic term}\rangle.$$

Here, ⟨linguistic modifier⟩ is an intensifying adverb with narrowing or extending effect. We use the following modifiers: *extremely, highly, very, rather, more or less, roughly, medium, quite roughly, very roughly, about, approximately* and the negation *not*. The ⟨atomic term⟩ may be an adjective (we use *small, medium, big*), a fuzzy number or some special term (e.g. *undefined*). A typical example of the linguistic expression $\mathcal{A}$ is *very small*, *roughly big* etc. There are some rules preventing certain combinations of modifiers and terms (e.g. *very medium* has no meaning) which are not important at this point. We described the level of linguistic descriptions. Below we describe the level of the meaning of the descriptions.

Each of the linguistic expressions is an example of the so called *evaluating expressions*, i.e., linguistic expressions, the meaning of which can be modelled by a fuzzy set on an ordered scale (usually on the set of real numbers). The fuzzy set representing the expression $\mathcal{A}$ is denoted by A. To get the meaning A, three facts have to be known. First, each variable involved has some *context* [1, 7]. The use of context is a very deep feature of natural language. The same word may have different meanings in different contexts (consider e.g. the word *small* in the context of everyday temperatures and in the context of air pressures). In our conception, the context is modeled by a real interval of possible values of the particular variable. Second, there is a rule assigning to every atomic term a subset in the context of the respective variable. For instance, for the adjectives *small*, *medium* and *big* we use the quadratic fuzzy sets of type S^-, Π and S^+, respectively [4]. Third, the meaning of linguistic modifiers is represented by a suitable modification of the fuzzy set which represents the atomic term to which the modifier is applied. A general theory of linguistic modifiers is presented in [5, 8, 9].

Note that the transparent relation between the level of syntax and semantics is kept even in adaptation of the whole model [1, 7, 10] due to the fact that the adaptation techniques reflect both of these levels.

To be able to derive decisions (control actions) we need a suitable inference method. Two kinds of methods are elaborated in our conception. As these methods are more or less known and are described elsewhere [2, 6] we give only the final formulas for the case of PI control.

---

[2]Note that in the case of the so called Takagi-Sugeno models the THEN-part of the rule has the form of an arithmetic expression.

[3]In fact, even more general form may be considered.

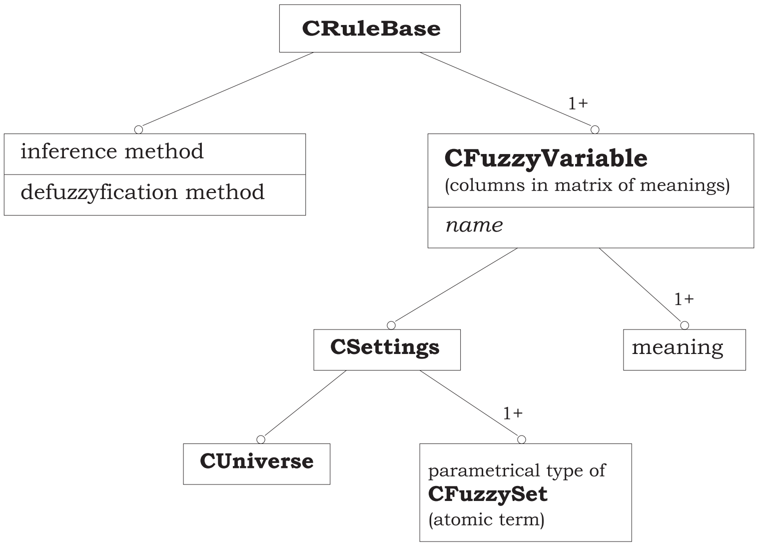

Figure 1: Structure of **CRuleBase**

Given the description of actual error and its change represented by the fuzzy sets A'^e, $A'^{\Delta e}$, the conclusion $B'^{\Delta u}$ may be obtained either by the Max–Min (Max–t–norm) rule:

$$B'^{\Delta u}y = \bigvee_{x_1,x_2} \Big((A'^e x_1 \wedge A'^{\Delta e} x_2) \wedge \bigvee_{i=1}^{n} \big((A_i^e x_1 \wedge A_i^{\Delta e} x_2) \wedge B_i^{\Delta u} y \big) \Big)$$

or by the fuzzy logic inference:

$$B'^{\Delta u}y = \bigvee_{x_1,x_2} \Big((A'^e x_1 \wedge A'^{\Delta e} x_2) \otimes \bigwedge_{i=1}^{n} \big((A_i^e x_1 \wedge A_i^{\Delta e} x_2) \rightarrow B_i^{\Delta u} y \big) \Big)$$

where $\otimes$ is the operation of Łukasiewicz product and $\rightarrow$ that of the corresponding residuation.

To derive a concrete control action from the output fuzzy set, a suitable defuzzification method has to be employed [3].

3 Design of LFLC

In this section we present a more technical description of the basic fuzzy and linguistic concepts briefly described in previous section. The analysis of the approximate reasoning leads us to realize that there are three most important notions, namely the fuzzy set, the linguistic settings and the linguistic description. Each of these notions has a corresponding counterpart in the implementation, namely the class. These classes are not at the same level of generality. Instances of the formers are members of the latters. Relationships among basic components of our design are shown in Figure 1.

3.1 Representation of fuzzy set

The first important class is **CFuzzySet**. Our intention was to make representation of fuzzy sets as general as possible. The purpose of this approach is to stress not only the properties of the fuzzy sets used in approximate reasoning. We were led to the creation of one fundamental class **CFuzzySet** which serves as the basis for all possible types of fuzzy sets.

All fuzzy sets have two common properties, namely the *context* and the *membership function*. **CFuzzySet** implements the context as a data member, being a multidimensional interval with special means for the most often used onedimensional case. The membership function is not included in **CFuzzySet**, but every type of fuzzy set (multidimensional, discrete, parametrically defined and fuzzy sets represented by means of α-cuts) has implementation of its membership function in the corresponding derived classes. Each of them has to implement the following methods declared in the base class:

- `GetMembDeg` — a general method for obtaining the membership degree from arbitrary membership function,
- `GetDiscretization` — a method which transforms arbitrary fuzzy set to discrete one

Types of fuzzy sets represented by classes derived from **CFuzzySet** are:

- discrete fuzzy sets (class **CFuzzyDiscreteSet**) which are widely used in the inference routines. They are, of course, also multidimensional with one-dimensional fuzzy sets as special case.
- parametrically defined families fuzzy sets (classes **CFuzzyTriangleSet**, **CFuzzyTrapezoidalSet** and **CFuzzyQudraticSet**) which are most often used as meanings of atomic linguistic terms.
- fuzzy sets represented by means of theier α-cuts (class **CFuzzyAlphaCutSet**).

Further, there several operations on fuzzy sets are implemented — intersection, union, implication, difference etc. These operations on fuzzy sets are realized using operations on real numbers, e.g. various kinds of t-norms and t-conorms. Thus, for example, intersection of fuzzy sets has no preferred hidden t-norm behind, but it is possible to use minimum, product, Łukasiewicz t-norm or other t-norms as well.

3.2 Representation of semantics of linguistic expressions

The main idea here is to permit the same linguistic expression to have different meanings dependingly on the user's choice of atomic terms and linguistic modifiers. We use the hierarchical structure of classes **C***Settings** with all the necessary information for obtaining the meaning of linguistic expressions which have the form described in Section 2. Base class **CSettings** contains only the information common for all types of semantics, namely the linguistic context and the virtual method `GetMeaning`, which returns a discrete fuzzy set — meaning of an actual linguistic expression. This method has to be implemented in all the derived classes. Derived classes have the following structure:

- **CBasicSettings** — base class for semantics which atomic terms are parametrically defined families of fuzzy sets:
  - **CLinguisticSettings** — atomic terms are fuzzy sets which quadratic membership functions,
  - **CTriangSettings** — triangular membership functions,
  - **CTrapezoidSettings** — trapezoidal membership functions,
- **CLinCSettings** — class representing the structure of succedent in Takagi-Sugeno fuzzy models.

Names and meanings of atomic terms and linguistic modifiers are placed as lists in the class **CBasicSettings**. There is also method `GetFuzzyConst` which returns the meaning of fuzzy number. The class **CLinguisticSetings** has special position among the family of **C***Settings** because atomic terms with quadratic meanings and linguistic modifiers based on shifting of the horizon have several specific properties. List of atomic terms placed in **CBasicSettings** serves here as list of user defined terms.

3.3 Representation of the linguistic descriptions

Our idea for representation of the linguistic description is to have one class which includes not only the information concerning the meanings of all the linguistic expressions appearing in the description, but also the structure of antecedent and inference and defuzzifications methods. In other words, all the data needed for the execution of inference should be here.

The information about meanings of the linguistic expressions is stored in a matrix. The rows of the matrix correspond to the particular rules of the linguistic description, the columns correspond to individual fuzzy variables and are represented by the class **CFuzzyVariable**. The most important data member of this class is an instance of **C***Settings**. It follows that, in general, every fuzzy variable can have different interpretation of the sematics of the linguistic expressions. Furthermore **CFuzzyVariable** contains the name of variable and its discretization[4] of the meanings of linguistic expressions. The column itself is the list where each member contains linguisic expression and the corresponding meaning. The meaning is usually fuzzy set but for fuzzy models of Takagi-Sugeno type it is an array of real numbers — coefficients of linear combination, which are necessary for an evaluation of arithmetical expression in the succedent part of rules.

4 Conclusion

The main goal of this contribution was to present theoretical background of fuzzy software systems and some principles used in its object-oriented implementation. Notions and concepts described are oriented mainly to the field of approximate reasoning, but all components of our system were designed with respect to easy applicability in other branches of soft computing.

References

[1] Bělohlávek R., Novák V.: Learning linguistic context for linguistic oriented fuzzy control. *Proc. of FUZZ-IEEE'97, vol.II*, 1167-1172, ISBN 0-7803-3796-4, Barcelona, 1997.

[2] Klawonn F., Novák V.: The Relation between Inference and Interpolation in the Framework of Fuzzy Systems. Fuzzy Sets and Systems 81(1996), 331–354.

[3] Klir G.J., Yuan B.: Fuzzy Sets and Fuzzy Logic. Theory and Apllications. Prentice Hall 1995.

[4] Novák V.: *Fuzzy Sets and Their Applications.* Adam–Hilger, Bristol, 1989.

[5] Novák V.: *The Alternative Mathematical Model of Linguistic Semantics and Pragmatics.* Plenum, New York, 1992.

[6] Novák V.: Fuzzy Control from The Logical Point of View. *Fuzzy Sets and Systems* **66**(1994), 159–173.

[7] Novák V.: Linguistically Oriented Fuzzy Logic Controller and Its Design. *Int. J. of Approximate Reasoning* **12**(1995), 263–277.

[8] Novák V.: A Transparent Model Of Linguistic Trichotomy and Hedges. *Proc. of EUFIT'96*, vol.1, 35-39, 1996.

[9] Novák V., Perfilieva I.: Evaluating Linguistic Expressions and Functional Fuzzy Theories in Fuzzy Logic. To appear in L.A. Zadeh and J. Kacpryk (eds.): Computing with Words in Systems Analysis. Springer-Verlag, Heidelberg 1998.

[10] Novák V., Smolíková R.: On learning of linguistic description from data in LFLC. *Proc. Int. Panel Conf. on Soft and Intelligent Computing.* Technical Univ. of Budapest, Budapest 1996, 229–234.

[11] Zadeh L. A.: Outline of a New Approach to the Analysis and Synthesis of Complex Systems and Decision Processes. IEEE Trans. System Man Cybernet., 1973, 28–44.

[12] Zadeh L. A.: The concept of a linguistic variable and its application to approximate reasoning I, II, III. *Inf. Sci.* **8**(1975), 199–257, 301–357; **9**(1975), 43–80.

[4]The number of points to which the membership function is sampled.