FOOM: A Fuzzy Object-Oriented Modeling for Imprecise Requirements

Jonathan Lee and Nien-Lin Xue
Software Engineering Lab.
Department of Computer Science and Information Engineering
National Central University
Chungli, Taiwan
{yjlee,nien}@se01.csie.ncu.edu.tw

Abstract— One of the foci of the recent development in object-oriented modeling (OOM) has been the extension of OOM to fuzzy logic to capture and analyze informal requirements that are imprecise in nature. In this paper, a novel approach to object-oriented modeling based on fuzzy logic is proposed to formulate imprecise requirements along four dimensions: (1) to extend a class by grouping objects with similar properties into a fuzzy class, (2) to encapsulate fuzzy rules in a fuzzy class to describe the relationship between attributes, (3) to evaluate the membership function of a fuzzy class by considering both static and dynamic properties, and (4) to model fuzzy associations between classes. The proposed approach is illustrated using the problem domain of a meeting scheduler system.

Keywords: Fuzzy logic, imprecise requirements, object-oriented modeling technique.

I. Introduction

One of the foci of the recent developments in object-oriented modeling (OOM) has been the extension of OOM to fuzzy logic to capture informal requirements that are imprecise in nature. Rumbaugh and his colleagues have argued that OOM is a way of thinking about problems using models organized around real-world concepts which are usually expressed in natural languages. As Zadeh pointed out in [14], it is evident that almost all concepts in or about natural languages are almost fuzzy in nature. Several researchers such as Dubois *et al.* [4], George *et al.* [6] and Lano [11] have further advocated that object classes with fuzzy memberships values are therefore a natural representation framework for real-world concepts.

As a continuation of our previous work [12] in using fuzzy logic as a basis for formulating imprecise requirements, we propose, in the paper, a fuzzy object-oriented modeling technique (FOOM) to capture and analyze imprecise requirements. In FOOM, we have identified several kinds of fuzziness that are required to model imprecise information involved in user requirements: (1) classes with imprecise boundary to describe a group of objects with similar attributes, similar operations and similar relationships; (2) rules with linguistic terms that are encapsulated in a class to describe the relationships between attributes; (3) ranges of an attribute with linguistic values or typical values in a class to define the set of allowed values that instances of that class may take for the attribute; (4) the membership degree (i.e. ISA degree) between an object and a class, and between a subclass and its superclass (i.e. AKO degree) can be mapped to the interval [0, 1]; and (5) associations between classes that an object instance may participate to some extent.

In the next section, different kinds of fuzziness rooted in FOOM are fully discussed in Section II. Related work is described in Section III. The benefits of our approach is summarized in Section IV. The notations used throughout the paper is an extension of Unified Modeling Language (UML).

II. Fuzzy Object-Oriented Modeling

A. Inside a Fuzzy Class

Traditionally, a class is used to describe a crisp set of objects with common attributes, common operations and common relationships. In order to model the impreciseness rooted in user requirements, we extend a class to describe a fuzzy set of objects (called a fuzzy class), in which objects may have similar attributes, similar operations and similar relationships, for example, a set of interesting books or a class of clever students. In the meeting scheduler system, the class *ImportantParticipant* is modeled as a fuzzy class, that is, a participant may be an important one to a degree.

A fuzzy class in FOOM is an encapsulation of a number of properties that can be classified as static properties or dynamic ones[1]. Static properties are viewed as integral features of an object that exist for its lifetime including identifier, attributes and operations. On the other hand, dynamic properties

[1] We have adopted this classification scheme from Zenith approach [5].

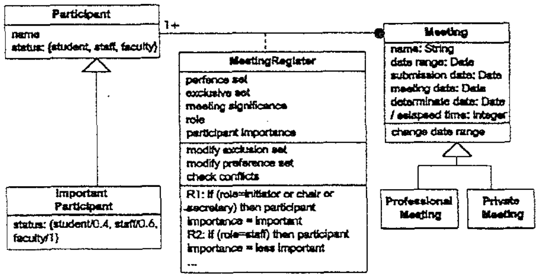

Fig. 1. An Example of a Fuzzy Class

are optional for an object and can be short-lived such as fuzzy rules[2] and fuzzy relationships.

Since a fuzzy class is a group of objects with similar static properties (i.e., attributes, operations) and similar dynamic properties (i.e., relationships and rules), the membership degree of an instance to a fuzzy class is dependent on the properties, especially the values of attributes and the values of link attributes. In our example, the degree that a person belongs to the class *ImportantParticipant* depends on his *status* and his *role* in the meeting he attends.

The *domain* of an attribute is the set of all values the attribute may take, irrespective of the class it falls into; whereas, the *range* of an attribute in a class is defined as the set of allowed values that a member of a class may take for the attribute. The range of an attribute a_i in the class C is denoted as $R(a_i, C)$. In FOOM, the fuzziness in the range of an attribute in a class may be due to either a linguistic term or a typical value.

- A class may be fuzzy for the linguistic values its attributes can take. For example, the class *YoungMan* has a fuzzy range for the attribute *age*, since a person may take *young* or *very young* as values for his age.
- The range of an attribute is fuzzy because some of its values are deemed as atypical (i.e. less possible than other values), therefore, each value the attribute may take is associated with a typical degree[3]. In our example, the class *ImportantParticipant* has a fuzzy range $\{student/0.4, staff/0.7, faculty/1\}$ for the attribute *status*, which means that a faculty is typically an important participant, and a student is an important participant with a typical degree of 0.4.

Using fuzzy rules is one way to deal with imprecision where a rule's conditional part and/or the conclusions part contains linguistic variables. Fuzzy rules are an optional feature for a fuzzy class in FOOM and are thus classified as a dynamic property. Fuzzy rules can be used to describe the internal relationship between attributes inside a class. For example, rules in Figure 1 describes the relationship between the attributes *role* and *participant importance*.

B. Fuzzy Classification

We extend crisp class memberships to fuzzy class memberships by allowing the existence of perceptual fuzziness. The notion of inheritance and its impact on the perceptual fuzziness are also elaborated.

B.1 Subclassing and Subtyping

Inheritance plays an important role in object-oriented analysis and design. Generally, inheritance is separated into two different concepts: implementation inheritance and subtyping inheritance. Implementation inheritance refers to the sharing of representation and implementation code between a subclass and its superclass. In contrast, subtyping refers to some kind of conformance between a subclass and its superclass in terms of their interfaces.

Subclassing is concerned with how a class is implemented, that is, a new class is constructed from a parent class by reusing some or all of the parent's operations [1]. In general, a subclass can be constructed through extension, redefinition and restriction. On the other hand, a type defines an abstract interface and can be viewed as the specification of objects behavior. In the specification level, the subtype-supertype relationship can be defined in terms of the weak form of the principle of substitutability.

Subclassing and subtyping are not necessarily related [1], namely, a subclass may be of different typing relationships: subtype, same type or supertype, to its superclass. In FOOM, a subclass is guaranteed to be either a subtype or a same type of its superclass, that is, the weak form of the principle of substitutability is maintained.

B.2 Relationship between Classes and Subclasses

As the weak form of substitutability is maintained in FOOM, a subclass is constructed through extending new operations, redefining the inherited operations, or modifying the inherited attribute ranges. In the first two cases, since an instance of a subclass possesses all properties that its superclass has, it is also an instance of its superclass, namely, the AKO degree is equal to 1. In the case of modifying the inherited attribute ranges, the attribute range in the superclass may include the attribute range in the subclass to some extent, that is, an instance of the subclass can be an instance of its superclass

[2] Encapsulating rules in an object is also proposed in SOMA [8].

[3] The notion of typical values is adopted from [4].

to a degree. Therefore, we will focus our attention on the case of modification for computing the AKO degree below.

The perceptual fuzziness is investigated by evaluating its static properties and dynamic properties. The criticality of an attribute indicates the relevance of the attribute to a perceptual fuzziness. For example, the attribute *participant importance* is more relevant than the attribute *status* while examining if a participant is an important one, therefore, it is assigned a higher criticality. To determine the criticality of attributes, we utilize the Analytic Hierarchy Process, which compares pairwise attributes according to their relative criticality. We use $CRI(a_i, C)$ to denote the criticality of an attribute a_i to the perceptual fuzziness for the fuzzy class C.

The membership degree of a class in its superclass has to account for the criticalities of the attributes and the degree of inclusion of the range of each attribute of the class in that of its superclass. Supposed that the class C is a superclass of the class D. The membership degree of the class C in the class D is denoted as $AKO(D, C)$, and defined as

$$AKO(D,C) = \sum_{a_i \in Att(C)} CRI(a_i, C) \times AKO_{a_i}(D,C) +$$

$$\sum_{E_k \in A(C)} \Big(\sum_{b_j \in Att(<C,E_k>)} CRI(b_j, C) \times AKO_{b_j}(D,C) \Big)$$

where $A(C)$ is the set of classes associated with C, $Att(C)$ is the set of attributes in C, and $Att(< C, E_k >)$ is the set of link attributes in the association $< C, E_k >$ which is established between the classes C and E_k. The degree $AKO_{a_i}(D, C)$ is the AKO degree with respect to (wrt) the attribute a_i and $AKO_{b_j}(D, C)$ refers to the AKO degree wrt to the link attribute b_j. Both static properties (object attributes) and dynamic properties (link attributes) are required for computing the membership degree of an object in a class.

To calculate $AKO_{a_i}(D, C)$, we need to examine whether the fuzziness of $R(a_i, C)$ is of the type of linguistic terms or typical values. In the case of linguistic terms, the membership degree is defined as the fuzzy inclusion of $R(a_i, D)$ into $R(a_i, C)$:

$$AKO_{a_i}(D,C) = INC_I(R(a_i, C)|R(a_i, D))$$

The degree of inclusion of fuzzy sets is defined as

$$INC_I(A|B) = \frac{||A \cap B||}{||B||}$$

In the case of typical values, the membership degree is defined by a fuzzy inclusion based on Godel's fuzzy implication:

$$AKO_{a_i}(D,C) = INC_G(R(a_i, C)|R(a_i, D))$$

The fuzzy inclusion $INC_G(A|B)$ between the fuzzy sets A and B is defined as:

$$INC_G(A|B) = inf_s(\mu_B(s) \rightarrow \mu_A(s))$$

Note that $INC(R(a_i, C)|R(a_i, D)) = 1$ if $\forall s, \mu_{R(a_i,D)}(s) \leq \mu_{R(a_i,C)}(s)$, that is, the range of the attribute a_i in the subclass is included in the corresponding range of its superclass. More specifically, the subclass inherits attributes from its parents by specializing their ranges.

Similarly, to calculate $AKO_{b_j}(D, C)$, we also need to examine whether the fuzziness of $R(b_j, < C, E_k >)$ is of the type of linguistic fuzziness or typical values. In the case of linguistic terms, the membership is defined as the fuzzy inclusion of $R(b_j, < D, E_k >)$ into $R(b_j, < C, E_k >)$:

$$AKO_{b_j}(D,C) = INC_I(R(b_j, < C, E_k >)|R(b_j, < D, E_k >))$$

In the case of typical values, the degree is defined as $INC_G(R(b_j, < C, E_k >)|R(b_j, < D, E_k >))$.

B.3 Relationship between Classes and Objects

The class membership between an object and a class is crisp, that is, the ISA degree of an object to a class is either 1 or 0. In FOOM, a perceptual fuzziness between an object and a class is allowed. An object may belong to a class to a degree. In the meeting scheduler system, a person may belong to the class *ImportantParticipant* to some extent.

The perceptual fuzziness of an object to a class is investigated by evaluating its static properties and dynamic properties. In our example, the membership degree of a person to the class *ImportantParticipant* can be obtained by checking his *status* and his *participant importance* in the meeting he attends.

The membership degree of an object x in a fuzzy class C is denoted as $ISA(x, C)$, and defined as:

$$ISA(x,C) = \sum_{a_i \in Att(C)} CRI(a_i, C) \times ISA_{a_i}(x,C) +$$

$$\sum_{E_k \in A(C)} \Big(\sum_{b_j \in Att(C,E_k)} CRI(b_j, C) \times ISA_{b_j}(x,C) \Big)$$

The degree $ISA_{a_i}(x, C)$ is the membership degree of the object x in the class C with respect to (wrt) the attribute a_i, and $ISA_{b_j}(x, C)$ refers to the membership degree wrt to the link attribute b_j.

To calculate $ISA_{a_i}(x, C)$, we also need to examine whether the fuzziness of $R(a_i, C)$ is of the type of linguistic terms or typical values. In the case of linguistic terms, the membership degree is defined as the degree of inclusion of the value of a_i of x into the range of a_i in C, that is,

$$ISA_{a_i}(x,C) = INC_I(R(a_i, C)|V(a_i, x))$$

where $V(a_i, x)$ is the value of x for a_i. In the case of typical values, the membership degree is defined as $\mu_{R(a_i,C)}V(a_i, x)$.

To calculate the membership degree wrt to a link attribute, we also need to check to see whether the

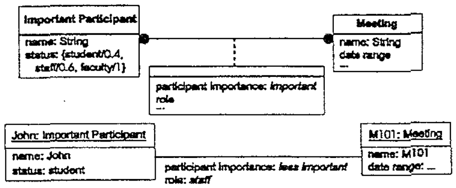

Fig. 2. An Example of Perceptual Fuzziness for ISA Relationship

fuzziness of the range of the link attribute's values is of the type of linguistic terms or typical values. In the case of linguistic terms, the membership degree is defined as the degree of inclusion of the value that b_j takes for the link $< x, e >$ into the range of b_j in $< C, E_k >$, where e is an instance of E_k connected with the object x:

$$ISA_{b_j}(x, C) = INC_I(R(a_i, < C, E_k >)|V(a_i, < x, e >))$$

In the case of typical values, the degree is defined as $\mu_{R(a_i, <C,E_k>)} V(a_i, x)$.

In our example, supposed that we have the following information in the meeting scheduler system:

$$R(status, IP) = \{student/0.4, staff/0.6, faculty/1.0\}$$
$$R(pi, < IP, Meeting >) = important$$
$$V(status, John) = student, V(role, John) = staff$$
$$V(pi, < John, M101 >) = less\ important$$
$$CRI(status, IP) = 0.3, CRI(pi, IP) = 0.7$$

where pi is the abbreviation of the attribute *participate importance*, and IP is the abbreviation of the class $ImportantParticipant$. To calculate the membership degree of a participant $John$ to the class IP (abbreviated as IP), we first calculate the membership degrees wrt the attributes *status* and *pi*.

$$ISA_{status}(John, IP) = \mu_{R(status,IP)}(student) = 0.4; \ and$$

$$ISA_{pi}(John, IP) = INC(important|less\ important) = 0.6$$

Therefore, we have $ISA(John, IP) = 0.54$. It should be noted that the importance degree of $John$ is dynamically determined by the meeting he attends. The example above describes that: "John is a more or less important participant (since the degree is 0.54) for the meeting M101".

C. Associations

In traditional object-oriented approaches, only crisp associations are introduced, namely, an object either participates in an association or not at all. An association A between classes C_1 and C_2 can be defined as a set of links:

$$A(C_1, C_2) = \{(< x, y >)|x \in C_1, y \in C_2\}$$

An association in FOOM is defined by a set of links and their corresponding *degree of participation* reflect the intensity that x and y participate in the association. Supposed that A is an association between class C_1 and C_2, we define

$$A(C_1, C_2) = \{((x, y), \mu_A(x, y))|x \in C_1, y \in C_2\}$$

where $\mu_A(x, y)$ is the degree of participation of the link $< x, y >$. In our example, if the degree of participation $\mu_{prefer(IP,Location)}(John, L102)$=0.9, it means that the degree John prefers the location $L102$ is 0.9.

III. Related Work

A number of researchers have reported progress towards the successful integration of fuzzy logic and object-oriented modeling, which can be classified into three categories based on their intended modeling purposes:

A. Knowledge representation for AI systems.

Lano has proposed to combine fuzzy reasoning and object-oriented representation for the real-world information [11]. A knowledge base is organized as a class hierarchy for representing concept categories, each class corresponds to a fuzzy set, whose membership functions is the proximity metric defined for the class. Learning is the process that transforms a knowledge base and a new example to a new knowledge base.

To support an approximate reasoning in systems based on prototypical knowledge representation, Torasso and Console have defined a formalism for the representations and a general evaluation mechanism to deal with the form of knowledge [13]. Each frame has three kinds of weighted attributes: necessary, sufficient and supplementary. The evaluation mechanism is based on fuzzy logic: the fuzzy match between prototypical description and sets of data is based on possibility theory and the relevance measure of each slot.

B. Data modeling for database systems.

In [4], Dubois and Prade have advocated that classes can be intensionally described in terms of attributes which are distinguished between the range of allowed values and the range of typical values. The degree of inclusion between a class C_1 and a subclass C_2 is computed by comparing the ranges or the typical ranges of C_1 with the ranges or the typical ranges of C_2. Three kinds of inheritance are proposed: typical inheritance, normal inheritance and atypical inheritance.

In [9], the problem of object recognition is viewed as a classification problem, which is characterized by an objected-oriented knowledge representation and control strategies based on fuzzy pattern matching procedures. Taxonomies of classes are represented by hierarchies of frames. Which class matching the unknown object is decided by fuzzy matching theory based on possibility theory. The matching process is seen as a quantification of similarity and differences of both objects, and the computation of these measures strongly depends on the types of the prototypes and object fields and the weight of each field.

Bodogna *et al.* [2],propose a Fuzzy Object Oriented model for management of crisp and fuzzy data. Their work develops a fuzzy graph-based data model, which intends to generalize a graph model so that imprecision and uncertainty can be managed at different levels. To formulate imprecise queries and retrieve precise or imprecise object with a degree of plausibility associated with them, fuzziness in managed in two level: imprecise in the data themselves and knowledge of the data, i.e., uncertainty in the information on the data.

C. Object-oriented modeling for conventional software systems.

George *et al.* have utilized the ranges of fuzzy values of classes and objects for computing the degree of inclusion and membership, respectively [6]. To measure the class memberships, a similarity metric is formulated to measure the nearness between attributes' values in a superclass and its subclasses.

Graham [7] has focused on the derivation of unknown values of attributes through the use of a-kind-of relation (AKO), generalized modus ponen and defuzzification techniques [7]. In Graham's work, the notion of an object is extended to that of a fuzzy object in two ways: (1) attributes' values may be fuzzy, and (2) AKO is a matter of degrees. The AKO degree between classes is assumed to be known by a system, and unknown attributes' values are derived through AKO, generalized modus ponen and defuzzification techniques.

In [10], the focus has been on the representation of uncertain information based on a generalized fuzzy sets notation. Gyseghem et al. represent fuzzy information as fuzzy sets and uncertainty by means of generalized fuzzy set. A generic class Fuzzy-Set is introduced to capture fuzziness associated with attributes. Uncertain information is modeled by a kind of generalized fuzzy sets in which each element of the universe is associated with a fuzzy truth value $\{p/true, n/false\}$.

However, none of these work is devoted to the development of an object-oriented technique for informal requirements, but either to the formulation of class memberships (e.g. [4], [7]), or to the representation of prototypical knowledge as in [13], [9]. Furthermore, features of object orientation are not fully explored in these work. Types of fuzziness in consideration are also somewhat limited.

IV. Conclusion

As was pointed by Borgida et al. [3], a good requirement modeling approach should take the problem of describing natural kinds into account; furthermore, Zadeh have indicated that almost all concepts in or about natural languages are almost fuzzy. In this paper, we have proposed an approach to incorporating fuzzy concepts into object-oriented systems for modeling imprecise requirements. Our approach offers two major benefits: (1) extending traditional object-oriented techniques by incorporating different kinds of fuzziness that are rooted in user requirements; and (2) providing a more flexible approach to evaluating the class memberships by taking both static and dynamic properties into account.

Acknowledge

This research was supported by National Science Council (Taiwan, R.O.C.) under grants NSC87-2213-E-008-019.

References

[1] B.S. Blair. *Object-Oriented Languages, Systems, and applications*. Pitman, 1991.

[2] G. Bordogna, D. Lucarella, and G. Pasi. A fuzzy object oriented data model. In *Proceedings of the Third IEEE Conference on Fuzzy Systems*, pages 313–318, 1994.

[3] A. Borgida, S. Greenspan, and J. Mylopoulos. Knowledge representation as the basis for requirements specification. *Computer*, pages 82–91, April 1985.

[4] D. Dubois, H. Prade, and J.P. Rossazza. Vagueness, typicality and uncertainty in class hierarchies. *International Journal of Intelligent Systems*, 6:161–183, 1991.

[5] Z.P. Kemp *et al.* Zenith system for object management in distributed multimedia design environments. *Information and Software Technology*, 34(7):427–436, 1992.

[6] R. George, R. Srikanth, F.E. Petry, and B.P. Buckles. Uncertainty management issue in the object-oriented data model. *IEEE Transitions on Fuzzy Systems*, 4(2):179–192, 1996.

[7] I. Graham. Fuzzy objects: Inheritance under uncertainty. In *Object Oriented Methods*, pages 403–433. Addison Wesley, 1994.

[8] I. Graham. Migration using SOMA: A semantically rich method of object-oriented analysis. *Journal of Object-Oriented Programming*, pages 31–42, February 1993.

[9] C. Granger. An application of possibility theory to object recognition. *Fuzzy Sets and Systems*, 28(3):351–362, 1988.

[10] N.V. Gyseghem, R.D. Caluwe, and R. Vandenberghe. UFO: Uncertainty and fuzziness in an object-oriented model. In *Proceedings of the Internationl Conference on Fuzzy Systems*, pages 773–778, 1993.

[11] K. Lano. Combining object-oriented representations of knowledge with proximity to conceptual prototypes. In *Proceedings of Computer Systems and Software Engineering*, pages 442–446, 1992.

[12] J. Lee and J.Y. Kuo. A new approach to requirements trade-off analysis for complex systems. *(to appear) IEEE Transactions on Knowledge and Data Engineering*, 1998.

[13] R.B. Terwilliger and R.H. Cambell. Please: Executable specifications for incremental software develoment. *The Journal of Systems Software*, 10:97–112, 1989.

[14] L.A. Zadeh. Test-score semantics as a basis for a computational approach to the representation of meaning. *Literacy Linguistic Computing*, 1:24–35, 1986.