УДК 004.8

Д.А. ТЕРЛЕЦКИЙ, А.И. ПРОВОТАР

НЕЧЕТКИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ДИНАМИЧЕСКИЕ СЕТИ. II[1]

Аннотация. Приведено обобщение объектно-ориентированных динамических сетей на нечеткий случай, которое позволяет представлять знания об объектах и классах объектов нечеткой природы, а также моделировать их изменения во времени. В рамках описанного подхода предложен механизм получения новых знаний на основе базовых, который значительно отличается от известных методов в существующих моделях представления знаний. Приведен пример построения конкретной нечеткой объектно-ориентированной динамической сети.

Ключевые слова: нечеткий объект, класс нечетких объектов, модификаторы, эксплуататоры.

ВВЕДЕНИЕ

В настоящее время одной из важных задач в области представления знаний является обобщение моделей представления знаний (МПЗ) на нечеткий случай с использованием такого мощного инструмента, как нечеткие множества [1]. Необходимость в таком обобщении обусловлена тем, что множество знаний, которыми обладает человек, в некотором смысле неточны, неполны или размыты [2, 3]. Поскольку нечеткие множества — эффективный аппарат для моделирования объектов и процессов такой природы, обобщения МПЗ и их использование целесообразны. Существует достаточно много подобных обобщений, в частности нечеткая логика, нечеткие семантические сети, нечеткие продукционные модели, нечеткие нейронные сети, нечеткие онтологии, нечеткие фреймы, нечеткий UML и т.п. Эти МПЗ, в первую очередь, являются теоретическими моделями, которые практически можно реализовать в рамках конкретных интеллектуальных информационных систем (ИИС) с применением той или иной парадигмы программирования. В последнее время наиболее популярно и широко используется объектно-ориентированное программирование (ООП). Большинство из известных МПЗ можно реализовать в конкретных ИИС с помощью ООП. Более того, такие МПЗ, как фреймы или скрипты, идеологически близки к нему. Данный подход также достаточно эффективен в построении и управлении базами данных [3, 4]. Однако при всех преимуществах в нем не предусмотрены возможности представления объектов нечеткой природы. Поэтому улучшение этого подхода с помощью нечетких множеств — актуальная задача [2, 3, 5].

Для представления нечетких и размытых знаний в данной статье излагается соответственное обобщение для такой МПЗ, как объектно-ориентированные динамические сети [6]. Для построения нечетких объектно-ориентированных динамических сетей используются понятия нечеткого объекта, класса нечетких объектов, а также операции над ними [7].

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРЕДСТАВЛЕНИЕ НЕЧЕТКИХ ЗНАНИЙ

Рассмотрим некоторые особенности объектно-ориентированного подхода к представлению нечетких знаний, основными понятиями которого являются нечеткие объекты, классы нечетких объектов и отношения между ними.

Каждый реальный или абстрактный объект имеет некоторые характеристические свойства, которые можно представить в виде атрибутов объекта-модели. Объект называется нечетким, если он имеет хотя бы одно нечеткое свойство, т.е. свойство, выраженное с помощью нечеткого множества [5, 8, 9].

[1] Начало см. в № 1, 2015.

Поскольку существуют объекты одинаковой природы, вводится понятие класса, в рамках которого они определены. Таким образом, каждый объект — конкретный пример некоторого класса (типа) объектов. В ООП класс рассматривается как некий прототип или абстрактный объект для построения конкретных объектов определенного типа. Класс объектов называется нечетким, если он имеет хотя бы одно нечеткое свойство [8, 9].

Класс объектов можно рассматривать с двух разных позиций: экстенсиональной и интенсиональной [10, 8, 9]. В первом случае он задается (формируется) с помощью определенного количества объектов, принадлежащих к нему, во втором — с помощью перечня его собственных атрибутов и их возможных значений. Может показаться, что оба способа представления класса объектов в конечном итоге сводятся к одному и тому же, несмотря на их различия, а именно к определенному классу объектов. На самом деле, классы объектов, полученные с помощью этих подходов, могут иметь принципиальные отличия.

Если использовать экстенсиональный подход, то полученный класс объектов будет зависеть от самих объектов, на основе которых он определяется. Если все атрибуты всех объектов определены, т.е. каждое свойство каждого объекта имеет конкретное значение или диапазон значений, то и класс, определенный на основе этих объектов, также будет определенным. В противном случае он будет частично определенным или неопределенным.

Если применять интенсиональный подход, то определенность полученного класса объектов будет зависеть от определения его атрибутов, т.е. если полностью определить все свойства класса объектов, то очевидно, что и сам класс будет определенным, в противном случае — частично определенным или неопределенным.

Важно также, кто и с какой целью определяет тот или иной класс объектов. Предположим, имеется три квадрата разных размеров. Если определять на их основе некий класс квадратов, то получим класс, описывающий квадраты только этих трех размеров. Если предположить, что нужно описать класс квадратов, при этом не имея в виду какие-либо конкретные квадраты, то теоретически можно определить класс квадратов, описывающий, например, квадраты из предыдущего случая. Если речь не идет о конкретных квадратах, то, скорее всего, определим класс, описывающий квадраты всех возможных размеров. Исходя из данного заключения можно сделать вывод, что оба способа представления классов объектов полезны, но при их использовании нужно учитывать указанные нюансы.

В ООП объекты и классы связаны с помощью четко определенных отношений в рамках соответствующей иерархии, построенной с помощью механизма наследования [11]. В случае нечетких объектов и классов нечетких объектов принадлежность объекта к классу может быть нечеткой (частичной), т.е. с некоторой мерой. Между нечеткими объектами и их классами выделяют отношения обобщения, агрегации и ассоциации [5, 9]. Отношение обобщения означает, что некоторый класс является подклассом или примером другого класса. Отношение агрегации показывает, что некоторый класс — составная часть другого класса. Отношение ассоциации отображает некие семантические отношения между классами, которые не связаны между собой отношением обобщения или агрегации.

В рамках объектно-ориентированного подхода к представлению нечетких знаний рассмотрим такую МПЗ, как фреймы, точнее, ее обобщение на нечеткий случай [12, 13]. Фреймы во многом похожи на ООП, но при этом во фреймовых системах акцент делается на инфраструктуре предметной области, построенной из объектов, классов и связей между ними, тогда как в ООП внимание акцентируется на обмене сообщениями между конкретными объектами [14]. Подобно классам и объектам в объектно-ориентированном подходе к представлению знаний фреймы также принято называть нечеткими, если они имеют хотя бы один нечеткий атрибут или частично наследуют другие фреймы [12]. Введение концепции частичного наследования в значительной мере меняет этот механизм, добавляя в него новые возможности.

Для построения нечетких объектно-ориентированных динамических сетей будем использовать концепцию объектно-ориентированных динамических сетей, описанную в [6], а также обобщения на нечеткий случай всех ее составляющих [7].

Определение 1. Нечеткая объектно-ориентированная динамическая сеть *FOODN* — это объектно-ориентированная динамическая сеть, для которой выполняется хотя бы одно из следующих условий:

1) $\exists A_k, \ldots, A_m \in O = \{A_1, \ldots, A_n\}$, где $1 \le k \le m \le n$ и $A_k, \ldots, A_m$ — нечеткие объекты;

2) $\exists T_p, \ldots, T_q \in C = \{T_1, \ldots, T_w\}$, где $1 \le p \le q \le w$ и $T_p, \ldots, T_q$ — классы нечетких объектов;

3) $\exists R_i, \ldots, R_j \in R = \{R_1, \ldots, R_v\}$, где $1 \le i \le j \le v$ и $R_i, \ldots, R_j$ — нечеткие отношения между нечеткими объектами и классами нечетких объектов.

В качестве отношений между нечеткими объектами и классами нечетких объектов, из которых состоит множество R, рассмотрим три упомянутых типа отношений в объектно-ориентированном подходе к представлению нечетких знаний. Отметим, что эти типы отношений в той или иной степени характерны для семантических сетей, фреймов, скриптов и детально описаны в соответствующей литературе, в частности в [14, 15].

Для иллюстрации *FOODN* используем пример *OODN*, приведенный в [6], поскольку он интуитивный и достаточно простой для понимания, а также отображает основные особенности предлагаемой МПЗ.

Пример 1. Рассмотрим классы нечетких объектов $T(Pg)$, $T(Rb)$ и $T(Sq)$, описывающие класс нечетких выпуклых многоугольников, класс нечетких ромбов и класс нечетких квадратов соответственно. Определим их спецификации и сигнатуры следующим образом:

$$T(Pg) = (P(Pg), F(Pg)) = (p_1(Pg), \ldots, p_4(Pg), f_1(Pg)),$$

$$T(Rb) = (P(Rb), F(Rb)) = (p_1(Rb), \ldots, p_6(Rb), f_1(Rb), f_2(Rb)),$$

$$T(Sq) = (P(Sq), F(Sq)) = (p_1(Sq), \ldots, p_6(Sq), f_1(Sq), f_2(Sq)).$$

Представим спецификации и сигнатуры классов нечетких объектов $T(Pg)$, $T(Rb)$, $T(Sq)$ с помощью табл. 1, 2.

Проанализировав табл. 1, заметим, что значение свойства p_2 указано как fuzzy, т.е. размеры сторон фигур представляются в виде нечетких множеств. Определение классов $T(Pg)$, $T(Rb)$ и $T(Sq)$ интенсиональное, поскольку в данной ситуации нет необходимости рассматривать классы, описывающие конкретные типы многоугольников, ромбов и квадратов. По этой же причине для свойств $p_4(Pg)$, $p_4(Rb)$ не указаны конкретные значения, а отмечены только диапазоны возможных значений.

Таблица 1. Спецификация классов нечетких объектов $T(Pg)$, $T(Rb)$, $T(Sq)$

Свойство, p_i	Семантическое значение свойств классов объектов	Значение свойств классов объектов		
		$T(Pg)$	$T(Rb)$	$T(Sq)$
p_1	Количество сторон	4	4	4
p_2	Размеры сторон	fuzzy	fuzzy	fuzzy
p_3	Количество углов	4	4	4
p_4	Градусные меры углов	$(0°, 180°)$	$(0°, 180°)$	$90°, 90°, 90°, 90°$
p_5	Равенство всех сторон	—	1	1
p_6	Равенство всех углов	—	fuzzy	1

Таблица 2. Сигнатура классов нечетких объектов $T(Pg)$, $T(Rb)$, $T(Sq)$

Метод, f_i	Функция исчисления	Тело метода классов объектов		
		$T(Pg)$	$T(Rb)$	$T(Sq)$
f_1	Периметр фигуры	$f_1(Pg)=\sum_{i=1}^{n} a_i$	$f_1(Rb)=4a$	$f_1(Sq)=4a$
f_2	Площадь фигуры	—	$f_2(Rb)=a^2\sin\alpha$	$f_2(Sq)=a^2$

Таблица 3. Спецификации нечетких объектов Rb_1 и Sq_1

Свойство, p_i	Значение свойств классов объектов	
	Ромб Rb_1	Квадрат Sq_1
p_1	4	4
p_2	({1.8 / 0.9 + 2 / 1 + 2.1 / 0.95}, см), ({1.8 / 0.9 + 2 / 1 + 2.1 / 0.95}, см), ({1.8 / 0.9 + 2 / 1 + 2.1 / 0.95}, см), ({1.8 / 0.9 + 2 / 1 + 2.1 / 0.95}, см)	({2.7 / 0.85 + 3 / 1 + 3.1 / 0.95}, см), ({2.7 / 0.85 + 3 / 1 + 3.1 / 0.95}, см), ({2.7 / 0.85 + 3 / 1 + 3.1 / 0.95}, см), ({2.7 / 0.85 + 3 / 1 + 3.1 / 0.95}, см)
p_3	4	4
p_4	95°, 85°, 95°, 85°	90°, 90°, 90°, 90°
p_5	1	1
p_6	0.8	1

Рассмотрим конкретные объекты классов $T(Rb)$ и $T(Sq)$, а именно нечеткий ромб Rb_1 и нечеткий квадрат Sq_1. Определим значения их свойств и методов, учитывая спецификации и сигнатуры их классов, с помощью табл. 3, 4 соответственно.

Таблица 4. Сигнатуры нечетких объектов Rb_1 и Sq_1

Метод, f_i	Тело метода классов объектов	
	Ромб Rb_1	Квадрат Sq_1
f_1	$f_1(Rb_1)=4a$	$f_1(Sq_1)=4a$
f_2	$f_2(Rb_1)=a^2\sin\alpha$	$f_2(Sq_1)=a^2$

Перейдем к непосредственному построению нечеткой объектно-ориентированной динамической сети для нечетких объектов Rb_1, Sq_1 и классов нечетких объектов $T(Pg)$, $T(Rb)$, $T(Sq)$. Очевидно, что множество нечетких объектов $O=\{Rb_1, Sq_1\}$, а множество классов нечетких объектов $C=\{T(Pg), T(Rb), T(Sq)\}$. Классы нечетких объектов $T(Rb)$, $T(Sq)$ — примеры класса нечетких объектов $P(Pg)$, а любой квадрат является ромбом. Согласно этим заключениям множество отношений R имеет следующий вид:

$$R=\{Rb_1 \xrightarrow{\text{instance-of}} T(Rb),\ Sq_1 \xrightarrow{\text{instance-of}} T(Sq),\ T(Rb) \xrightarrow{\text{a-kind-of}} T(Pg),$$
$$T(Sq) \xrightarrow{\text{a-kind-of}} T(Pg),\ T(Sq) \xrightarrow{\text{is-a}} T(Rb)\},$$

что также можно записать как

$$R=\{Rb_1 \in T(Rb), Sq_1 \in T(Sq), T(Pg)\subseteq T(Rb),\ T(Pg)\subseteq T(Sq),\ T(Rb)\subseteq T(Sq)\}.$$

Определим множество эксплуататоров E, используя универсальные операции над объектами и классами, предложенные в [16]; их практическое применение к нечетким объектам и классам нечетких объектов рассмотрено в [7]. Таким образом, множество эксплуататоров E состоит из операций объединения, пересечения, разности, симметрической разности, а также операции клонирования нечетких объектов и классов нечетких объектов, т.е.

$$E=\{E_1^n, E_2^n, E_3^2, E_4^2, E_5^1\}=\{\cup, \cap, \setminus, \div, Clone_i\},$$

где E_1^n — операция объединения, E_2^n — операция пересечения, E_3^2 — операция разности, E_4^2 — операция симметрической разности, E_5^1 — операция клонирования. Нижний индекс каждого эксплуататора — его номер в сигнатуре, а верхний — арность. Эти операции в некоторой степени универсальны, поскольку могут применяться ко всем нечетким объектам и их классам.

Определим множество модификаторов M следующим образом:

$$M = \{M_1(T(Sq)) : T(Sq) \to T(Rb),\ M_1(T(Rb)) : T(Rb) \to T(L_1),$$
$$M_2(T(Rb)) : T(Rb) \to T(Sq),\ M_1(T(Pg)) : T(Pg) \to T(L),\ M_1(Sq_1) : Sq_1 \to Rb_1,$$
$$M_1(Rb_1) : Rb_1 \to L_{1_1},\ M_2(Rb_1) : Rb_1 \to Sq_1\},$$

где $M_1(T(Sq)) = M_1(P(Sq)) = M_1(p_6(Sq)) = M_1(1) = 0.8$ — частичный модификатор, который изменяет свойство $p_6(Sq)$ класса нечетких квадратов $T(Sq)$, тем самым трансформируя его в класс нечетких ромбов $T(Rb)$; $M_1(T(Rb)) = M_1(p_1(Rb)) = M_1(4, \textit{стороны}) = (3, \textit{отрезка})$ — частичный модификатор, который изменяет свойство $p_1(Rb)$ класса нечетких ромбов $T(Rb)$, трансформируя его в некоторый класс нечетких ломаных линий $T(L_1)$; $M_2(T(Rb)) = M_2(P(Rb)) = M_2(p_6(Rb)) = M_2(0.8) = 1$ — частичный модификатор, который изменяет свойство $p_6(Rb)$ класса нечетких ромбов $T(Rb)$, трансформируя его в класс нечетких квадратов $T(Sq)$; $M_1(T(Pg)) = M_1(p_1(Pg)) = M_1(4, \textit{стороны}) = (3, \textit{отрезка})$ — частичный модификатор, который изменяет свойство $p_1(Pg)$ класса нечетких выпуклых многоугольников $T(Pg)$, трансформируя его в некий класс нечетких ломаных линий $T(L)$; $M_1(Sq_1) = M_1(p_4(Sq_1), p_6(Sq_1)) = M_1((90°, 90°, 90°, 90°), 1) = ((95°, 85°, 95°, 85°), 0.8)$ — частичный модификатор, который изменяет свойства $p_4(Sq_1)$ и $p_6(Sq_1)$, трансформируя нечеткий квадрат Sq_1 в нечеткий ромб Rb_1; $M_1(Rb_1) = M_1(p_1(Rb_1)) = M_1(4, \textit{стороны}) = (3, \textit{отрезка})$ — частичный модификатор, который изменяет свойство $p_1(Rb_1)$ нечеткого ромба Rb_1, трансформируя его в некую нечеткую ломаную линию L_{1_1}; $M_2(Rb_1) = M_2(p_4(Rb_1), p_6(Rb_1)) = M_2((95°, 85°, 95°, 85°), 0.8) = ((90°, 90°, 90°, 90°), 1)$ — частичный модификатор, который изменяет свойства $p_4(Rb_1)$ и $p_6(Rb_1)$ нечеткого ромба Rb_1, трансформируя его в нечеткий квадрат Sq_1.

Проанализировав модификаторы из множества M, заметим, что некоторые из них модифицируют одновременно несколько свойств нечетких объектов и классов нечетких объектов. Свойства, которые модифицируются, выбраны не случайно, поскольку все они тем или иным образом связаны и изменение одного из них должно повлечь соответственные изменения остальных. Это следует из принципа рефлексии, рассмотренного в [7].

Для упрощения, а также понимания структуры и сути нечеткой объектно-ориентированной динамической сети, построенной для объектов Rb_1, Sq_1 и классов объектов $T(Rb)$, $T(Sq)$, $T(Pg)$, представим ее в виде связного ориентированного графа (рис. 1).

Вершинами графа считаем нечеткие объекты Rb_1, Sq_1, L_{1_1}, L_{2_1} и классы нечетких объектов $T(Rb)$, $T(Sq)$, $T(Pg)$, $T(L)$, $T(L_1)$, $T(L_2)$, а связями между ними — отношения a-kind-of, instance-of, is-a и модификации $M_1(T(Sq))$, $M_1(T(Rb))$, $M_2(T(Rb))$, $M_1(T(Pg))$, $M_1(Rb_1)$, $M_2(Rb_1)$, $M_1(Sq_1)$. Отношения в некоторой степени связывают между собой объекты и классы объектов, формируя при этом определенную иерархию. Эта часть сети статическая, поскольку с ее помощью можно представить только структуру знаний о некоторых объектах и классах объектов.

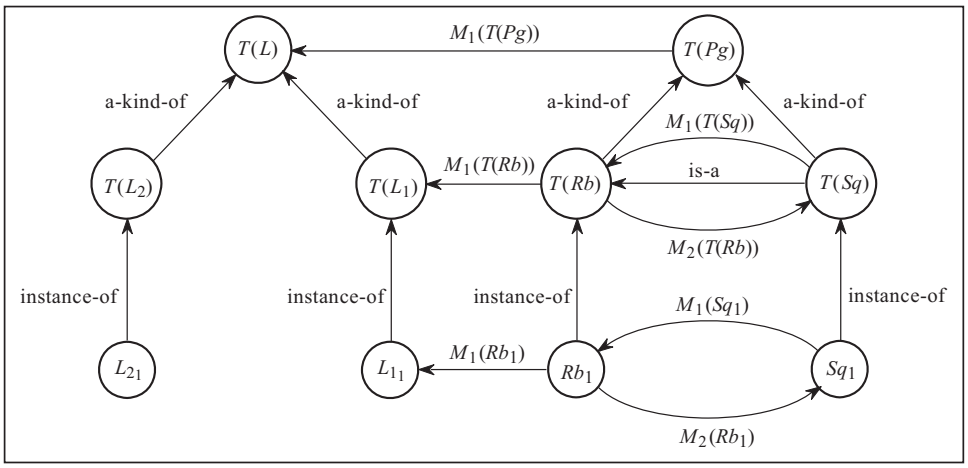

Рис. 1. Фрагмент *FOODN* для нечетких объектов Rb_1, Sq_1 и классов нечетких объектов $T(Rb)$, $T(Sq)$ и $T(Pg)$

Модификации можно рассматривать как некий новый тип отношений между объектами и классами объектов, представляемый как modification-of. Исходя из рис. 1, отметим, что $T(Sq) \xrightarrow{\text{modification-of}} T(Rb)$ и, наоборот, $T(Rb) \xrightarrow{\text{modification-of}} T(Sq)$. Но, в отличие от остальных типов отношений, модификации не полностью статические, поскольку сами модификаторы являются некоторыми методами, которые можно применить к объектам или классам объектов, при этом изменяя их.

Если речь идет о модификации некоторого нечеткого квадрата Sq_k с помощью некоторого модификатора $M_1(Sq_k)$, который трансформирует его в некий нечеткий ромб Rb_m, то в результате из нечеткого квадрата получим нечеткий ромб, а не то и другое одновременно, как это показано на рис. 1 на примере модификации нечеткого квадрата Sq_1. Приведенная на рис. 1 *FOODN* построена на основе объектов Sq_1 и Rb_1, поэтому они являются составляющими частями сети. Такой способ графического представления процесса модификации был выбран с целью показать изменения сущности объекта или класса объектов после применения к нему модификатора. Это решение может показаться не самым оптимальным, но если воспринимать модификацию объекта как процесс создания нового класса объектов, а не только как изменение конкретного объекта, то выбранный способ графического представления процесса модификации достаточно обоснованный.

На рис. 1 показан фрагмент нечеткой объектно-ориентированной динамической сети без изображения эксплуататоров. На рис. 2 приведены принципы действия эксплуататоров построенной *FOODN*.

На рис. 2 представлен граф, который является частью *FOODN*, изображенной на рис. 1. В качестве вершин графа выступают классы нечетких объектов. Основное значение имеют классы $T(Rb)$ и $T(Sq)$, поскольку они являются аргументами для пяти типов эксплуататоров из множества эксплуататоров E. Все остальные классы объектов — результаты использования эксплуататорами E_1^n, E_2^n, E_3^2, E_4^2, E_5^1 классов $T(Rb)$ и $T(Sq)$. Для изображения результата каждой эксплуатации используется определенное количество ребер, которое равно арности соответствующего эксплуататора. Например, для представления результата эксплуатации классов объектов $T(Rb)$ и $T(Sq)$ эксплуататором E_1^2 используются два ребра: $\cup T(Rb)$ и $\cup T(Sq)$. Часть ребер графа отмечена штриховой линий, чтобы подчеркнуть факт, что пересечение, разность и симметрическая разность объектов или классов объектов не всегда существуют [16]. Заметим, что рис. 2 иллюстрирует результаты использования эксплуататорами из множества E только двух классов нечетких объектов из множества C. Аналогичные графы можно построить для

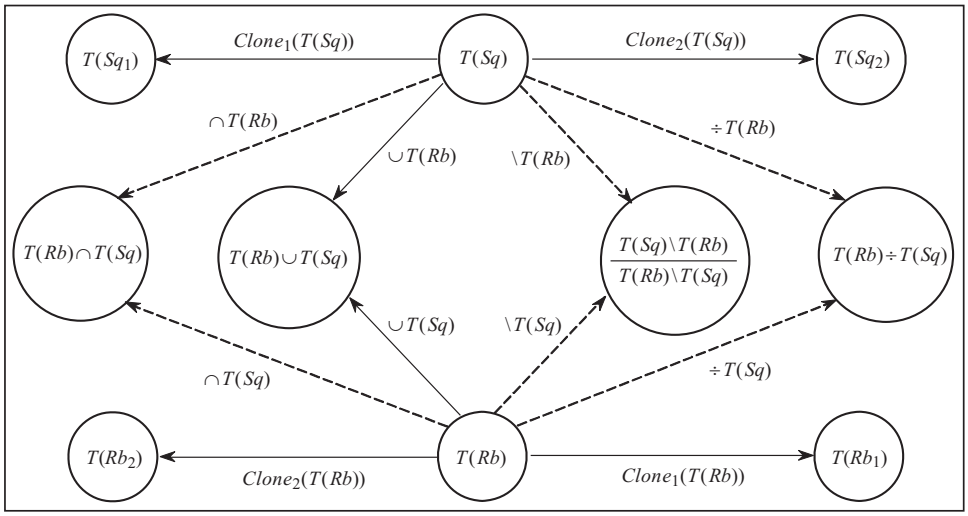

Рис. 2. Графическое изображение принципов действия эксплуататоров для множеств объектов и классов объектов

любых нечетких объектов из множества O и классов нечетких объектов из множества C. Также можно применять эксплуататоры к нечетким объектам или классам нечетких объектов, полученных в результате предыдущего использования эксплуататоров, порождая новые объекты и классы объектов.

Модификаторы и эксплуататоры формируют динамическую составляющую сети, поскольку с их помощью можно расширять и модифицировать сеть, таким образом получая новые знания, а также моделировать их изменения во времени.

В представленном примере в качестве эксплуататоров рассматривались только универсальные операции над объектами и классами объектов. Отметим, что множество эксплуататоров E, так же как и множество модификаторов M, может состоять из любых методов, которые удовлетворяют соответствующим определениям эксплуататора и модификатора, приведенным в [6].

Эксплуататоры имеют важное значение в объектно-ориентированных динамических сетях, поскольку их использование позволяет создать новые объекты и классы объектов на основе базовых объектов и классов объектов, на которых основана сеть. Таким образом, эксплуататоры дают возможность получить знания, которые неочевидны, и тем самым расширить описание той или иной предметной области. Результаты возможных применений эксплуататоров к объектам или классам объектов приведены в табл. 5.

Анализируя табл. 5, можно видеть, что операции пересечения, разности и симметрической разности позволяют генерировать новые классы объектов, а операция клонирования — копии объектов или классов объектов. Особый интерес вызывает операция объединения, которая дает возможность создавать множества объектов и классы объектов на основе элементов этих множеств. Данный подход к созданию классов объектов по своей сути экстенсионален, но при этом он позволяет создавать неоднородные классы, т.е. классы, описывающие объекты разных типов. Такие классы имеют непосредственное отношение к множествам объектов, которые могут состоять не только из однотипных элементов. В данном

Таблица 5. Результаты различных применений эксплуататоров E_1^n, E_2^n, E_3^2, E_4^2, E_5^1

Эксплуататор	Результаты применения к объектам	Результаты применения к классам объектов
$\cup$	Множество объектов и класс объектов	Класс объектов
$\cap$	Класс объектов	
$\setminus$		
$\div$		
$Clone_i$	Объект	

случае класс объектов, формирующий множество объектов, является неким прототипом, несмотря на свою экстенсиональную природу, поскольку, определив такой класс, можно строить новые объекты этого типа. Отметим, что в его рамках можно построить только объекты, эквивалентные тем, что принадлежат множеству объектов, на основе которого определен этот класс. Во многих существующих МПЗ основными репрезентативными составляющими являются объекты, классы и концепты, которые они репрезентируют. Если использовать множества объектов для представления знаний, то, в отличие от объектов, классов и даже концептов, они позволяют рассматривать одновременно некое количество объектов, причем не обязательно одного типа, что дает возможность описывать более сложные структуры знаний.

ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена концепция объектно-ориентированного подхода к представлению нечетких знаний. Предложено обобщение объектно-ориентированных динамических сетей на нечеткий случай. Продемонстрировано представление нечетких знаний на примере построения нечеткой объектно-ориентированной динамической сети для некоторых классов нечетких выпуклых многоугольников. Приведенный подход позволяет представлять нечеткие знания, моделировать их изменения во времени, а также предоставляет механизм получения новых знаний на основе базовых, который значительно отличается от известных методов получения знаний в существующих МПЗ.

СПИСОК ЛИТЕРАТУРЫ

1. Zadeh L.A. Fuzzy sets // Information and Control. — 1965. — **8**, N 3. — P. 338–353.
2. Leung K.S., Wong M.H. Fuzzy concepts in an object oriented expert system shell // International Journal of Intelligent Systems. — 1992. — **7**, N 2. — P. 171–192.
3. Berzal F., Marin N., Pons O., Vila M.A. Managing fuzziness on conventional object-oriented platforms // International Journal of Intelligent Systems. — 2007. — **22**, N 7. — P. 781–803.
4. Marin N., Pons O., Vila M.A. Fuzzy types: a new concept of type for managing vague structures // International Journal of Intelligent Systems. — 2000. — **15**, N 11. — P. 1061–1085.
5. Ndousse T.D. Intelligent systems modeling with reusable fuzzy objects // International Journal of Intelligent Systems. — 1997. — **12**, N 2. —— P. 137–152.
6. Terletskyi D.O., Provotar A.I. Object-oriented dynamic networks // Computational models for business and engineering domains / Ed. by G. Setlak, K. Markov. — Rzeszow: ITHEA, 2014. — 298 p.
7. Terletskyi D.A., Provotar A.I. Fuzzy object-oriented dynamic networks. I // Cybernetics and Systems Analysis. — 2015. — **51**, N 1. — P. 34–40.
8. Ma Z.M., Zhang W.J., Ma W.Y. Extending object-oriented databases for fuzzy information modeling // Journal Information Systems — Databases: Creation, Management and Utilization. — 2003. — **29**, N 5. — P. 421–435.
9. Zhang F., Ma Z.M. Construction of fuzzy ontologies from fuzzy UML models // International Journal of Computational Intelligence Systems. — 2013. — **6**, N 3. — P. 442–472.
10. Bordogna G., Pasi G., Lucarella D. A fuzzy object-oriented data model for managing vague and uncertain information // International Journal of Intelligent Systems. — 1999. — **14**, N 7. — P. 623–651.
11. Stroustrup B. The C++ programming language: Fourth edition. — Upper Saddle River (N.J.): Addison-Wesley Professional, 2013. — 1368 p.
12. Graham I., Jones P.L. A theory of fuzzy frames: part 1 // Bulletin for Studies and Exchanges on Fuzziness and its Applications. — 1987. — N 31. — P. 109–132.
13. Graham I., Jones P.L. A theory of fuzzy frames: part 2 // Bulletin for Studies and Exchanges on Fuzziness and its Applications. — 1987. — N 32. — P. 120–135.
14. Brachman R.J., Levesque H.J. Knowledge representation and reasoning. — San Francisco: Morgan Kaufmann, 2004. — 381 p.
15. Negnevitsky M. Artificial intelligence: A guide to intelligent systems. — 2nd ed. — Harlow: Addison-Wesley, 2004. — 440 p.
16. Terletskyi D.O., Provotar O.I. Mathematical foundations for designing and development of intelligent systems of information analysis // Problems in Programing. — 2014. — N 2–3. — P. 233–241.

Поступила 03.06.2015