УДК 681.3.06

НЕЧЕТКИЕ МОДЕЛИ МУЛЬТИАГЕНТНЫХ СИСТЕМ В РАСПРЕДЕЛЕННОЙ СРЕДЕ

И.Н. Парасюк, С.В. Ершов

Институт кибернетики имени В.М. Глушкова НАН Украины,
03680, Киев-187, проспект Академика Глушкова, 40, тел. 526 6422, e-mail: ivpar1@i.com.ua

Рассмотрены архитектурные нечеткие модели мультиагентных систем в распределенной среде, обладающие преимуществами как реактивных, так и мотивированных делиберативных агентов. Предложены средства спецификации поведения нечетких агентов на основе трансформаций нечетких графов с использованием нечеткого координатора и механизма адаптации нечетких правил. Формализована процедура согласования поведения нечетких агентов в мультиагентных системах относительно нечетких атрибутов, задающих свойства объектов. Предложена функция полезности, позволяющая получить оптимальное поведение нечетких агентов в процессе многоатрибутного итеративного согласования в распределенной среде.

Architectural fuzzy models of multiagent systems in distributed environment that possesses advantages of both reactive and motivated deliberative agents, are considered. Means of behavior specification of fuzzy agents are offered on the basis of fuzzy graphs transformations using fuzzy coordinator and mechanism for adaptation of fuzzy rules. A negotiation procedure of fuzzy agents' behavior in multiagent systems is formalized in relation to fuzzy attributes that define properties of objects. A fitness function that allows to achieve an optimum behavior of fuzzy agents in the process of multiattribute iterative concordance in distributed environment, is offered.

Современная теория и практика программирования в настоящее время все чаще исследует модели построения интеллектуальных мультиагентных систем, образованных взаимодействующими агентами [1, 2]. Оригинальные результаты получены, например, в работах [3,4] по инсерционной технологии программирования и моделирования (с введением агентов, обладающих недетерминированным поведением при погружении в среду, которую сами меняют).

Нечеткие модели по замыслу Л.А. Заде [5] должны использоваться при построении методов и средств компьютеризации гуманистических систем, в основу которых положен принцип существенного гранулирования информации. Значительному расширению области применения нечетких мультиагентных систем препятствует недостаточный уровень формализации концептуальных нечетких моделей агентов, отсутствие средств координации и согласования нечетких показателей при совместном решении задач агентами в распределенной среде.

Целью данной работы является формализация поведения интеллектуальных агентов на основе нечетких моделей и разработка процедуры согласования поведения агентов в нечетких мультиагентных системах. Данная статья является дальнейшим развитием исследований в направлении становления моделеориентованных архитектур программных систем в нечетком представлении [6-8].

Архитектура интеллектуальных агентов: общие положения

Под термином интеллектуальный агент понимаются программы, получающие информацию из окружающей среды и выполняющие над ней соответствующие операции, при этом их поведение рационально[9]. Архитектуры таких агентов делятся на четыре общих класса: дедуктивные агенты, реактивные агенты (агенты, основанные на поведении), делиберативные ("разумные") агенты и агенты, основанные на побуждениях [1,10-13].

При дедуктивном подходе агент манипулирует символьным представлением среды, используя операции логического вывода [1,10,11]. Агенты играют роль программ автоматического доказательства теорем, непосредственно выполняя логические спецификации. Основные недостатки данного подхода: проблема трансдукции, необходимость обеспечения непротиворечивых, надежных и "четких" знаний, невозможность манипулировать описанием динамической и стохастической среды, чрезвычайная иерархичность, необходимость построения полного агента для проведения испытаний, что не позволяет проводить итерационные пошаговые процессы разработки.

Парадигма Убеждение-Желание-Намерение (Belief-Desire-Intention) воплощает один из основных видов делиберативных ("разумных" агентов) и содержит явно представленную структуру данных, соответствует этим трем указанным свойствам рассуждений, применяемых агентом при решении задач [1,10,11].

В агенте BDI [1,10], среда E начинается в специфическом состоянии в конечного множества дискретных состояний: $E = \{e, e', ...\}$. Агент Ag выбирает специфическое действие из доступного множества возможных действий A на основе состояния среды и предыдущих выполненных действий: $A=\{\alpha,\alpha',\ldots\}$. Запуск R - последовательность чередующихся состояний и действий. Среда запускается в определенном состоянии и агент выбирает действие, чтобы выполняться при этом состоянии. В результате этого действия, среда может перейти в ряд возможных состояний:

$$R = e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{u-1}} e_u .$$

Функция *Ag* отображает запуски на действия, агент выбирает следующее действие, основываясь на доступной истории запуска системы: $Ag : R \to A$. Агент пробует найти и выполнить лучший план $plan(\pi)$, который является серией действий, выполнимых в определенном порядке: $\pi = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

BDI агент выполняет следующие функции:

1. Функция пересмотра убеждений агента (т.е. *brf*) модифицирует текущие убеждения агента на основе множестве всех убеждений *Bel* и текущих результатов перцепции: $brf : \wp(Bel) \times Per \to \wp(Bel)$.

2. Желание, которое агент стремится достичь, считывается из стека *намерений*. Этот стек содержит все *желания* (т. е. *цели*), которые агент обязался выполнить. Агент отыскивает *планы*, указывающие желание, извлеченное из стека, в качестве постусловий. Из этих планов, только те, которые удовлетворяют своему предусловию (на основе его текущих убеждений), становятся возможными *вариантами* или *желаниями* для агента.

Делиберативный агент начинает *размышлять*, используя два функциональных компонента: функция *выбора* $options : \wp(Bel) \times \wp(Int) \to \wp(Des)$, которая для текущего множества убеждений и намерений *Int* агента и производит множество возможных желаний *Des*. Для того, чтобы выбрать между конкурирующими вариантами, агент использует функцию *фильтрации* $filter : \wp(Bel) \times \wp(Des) \times \wp(Int) \to \wp(Bel)$. В процессе рассуждений дальнейшие желаний помещаются в стек намерений, что в свою очередь запускает поиск большего количества планов, чтобы достичь указанной цели, и так далее. Процесс заканчивается отдельными действиями, которые могут быть непосредственно выполнены. Если специфический план достижения цели терпит неудачу, то агент может выбрать другой план достижения желания из всех возможных планов.

Так как размер структуры намерений становится очень большим, ограничения, накладываемые на вычислительные ресурсы для планирования, уточнения и оценки этих намерений, вызывают “познавательную перегрузку” агента BDI. К сожалению, ранее упомянутые механизмы фильтрации, не помогают значительно сократить “познавательную перегрузку” поскольку, со временем генерируется ряд действительных альтернатив, которые все совместимы с текущими намерениями агента.

Другим возможным решением является использование реактивных (поведенческих) агентов [1,10-12]. В таких агентах не рассматриваются планирующие действия, основанные на сложных внутренних представлениях окружающей среды из-за их неотъемлемых ошибок их четкого представления и ассоциированных с их обработкой временных затрат. Реактивный агент Ag_r производит ответ из множества возможных ответов $R = \{r_1, r_2, \ldots\}$, на основе текущих стимулов *S*, что может быть представлено следующим отображением: $Ag_r : \beta(S) \to R$. Каждый индивидуальный стимул или результат перцепции s_i (где $s_i \in S$) - кортеж, состоящий из специфического типа или перцепционного класса *p* и значения силы λ: $s_i = (p, \lambda)$. Агент Ag_r выбирает текущий ответ *r*) из множества возможных доступных ответов *R* каждый раз когда λ больше, чем порог τ. Функция поведения β, ответственная за отображение входных стимулов на множество последовательной действий, может быть дискретной или непрерывной. Реактивные агенты могут быть разработаны с использованием итеративного процесса с использованием кооперативных или конкурентных методов композиции отдельно задаваемых функций поведения β_i (где $\beta_i \in \beta$). Недостатком таких методов композиции является их сложность. Так, в архитектуре реактивного агента если задано *n* уровней поведения, и каждый уровень способен предложить *m* возможных действий, то это означает, что необходимо задать m^n их взаимодействий. Кроме того, стремясь упростить функцию поведения, такие агенты не учитывают возможного состояния самого агента, его состояния и дальнейшей мотивации.

В качестве возможной альтернативы можно рассматривать мотивированные агенты [13] как подход, совместимый с идеями делиберативных и поведенческих агентов. Различные побуждения (мотивации) - ключевые определяющие факторы, с помощью которых агент может производить разнообразное поведение, имеющее высшую степень выгоды (или полезности) как для агента, так и для мультиагентной системы в целом.

При таком подходе, мотив *m* – это отображение текущих убеждений *bel* агента о состоянии его среды во множество активных побуждений $m : \wp(bel) \to \wp(m)$. Такой агент будет иметь две функции: порождение цели и активация цели обе из которых предназначены для генерирации активных целей в ответ на обнаруженные изменения в его текущих убеждениях. Порожденные цели добавляются ко множеству активных целей агентов: $gen : \wp(m) \times \wp(bel) \to \wp(mg)$. Активация (вызов) цели происходит, когда интенсивность побуждения связанного с целью превышает определенный *th*-порог: $act : \wp(mg) \times \wp(bel) \times \wp(th) \to \wp(goal)$.

Привлекательность этого подхода по сравнению с реактивными агентами в том, что пользовательские ожидания относительно поведения агента отображаются в профиль побуждений, который может затем быть сопоставлен с реальным поведением агента и усовершенствован. Кроме того, в отличие от делиберативного метода, такой тип агентов не использует сложную систему активации целей и с ней фильтры или триггеры.

Концептуальные основы построения нечетких мультиагентных систем в распределенной среде

Наиболее простой тип нечетких агентов – реактивные агенты с простым поведением, модель которых основана только на текущем состоянии среды. Их агентская функция основана на схеме условие-действие: IF (условие) THEN действие.

Нечеткий логический вывод в таких агентах выполняется по нечеткой базе знаний:

$$\bigcup_p \left(\bigcap_i x_i = a_{ijp} \text{ с весом } w_{jp} \right) \to \beta = d_j ,$$

где $x_i, \beta, i = \overline{1,n}$ – лингвистические переменные, а a_{ijp}, d_j, $j = \overline{1,m}$, $p = \overline{1,k}$ – лингвистические термы. Поскольку для выполнения агентом действия часто необходима дефузификация нечетких значений, логический вывод осуществляется как по алгоритму Мамдани, так и по алгоритму Сугено, либо с использованием иерархической системы нечеткого вывода [14].

Рассмотрим спецификацию поведения нечетких агентов в мультиагентной системе, предназначенной для закупки (продажи) товаров и услуг в распределенной среде. Система состоит из набора агентов-поставщиков, агентов-потребителей, представляющих интересы участников рынка, и агента-диспетчера, предназначенного для координации запросов.

Агенты вычисляют значения своих функций поставки и потребления, используя знания, представленные в форме нечетких правил. Например, цена товара является лингвистической переменной со значениями *низкая* (A_1) и *высокая* (A_2), а потребность в товаре (его необходимое количество) – лингвистической переменной со значениями *малая* (B_1) та *большая* (B_2) (рис.1).

1. Агент-потребитель содержит четыре (2*2) нечетких правила, позволяющих определить значение лингвистической переменной спроса. Например:

ЕСЛИ Цена=A1 И Потребность=B1 ТО Количество=C2(среднее).

2. Агент-поставщик также включает четыре нечетких правила, на основе которых определяются значения спроса.

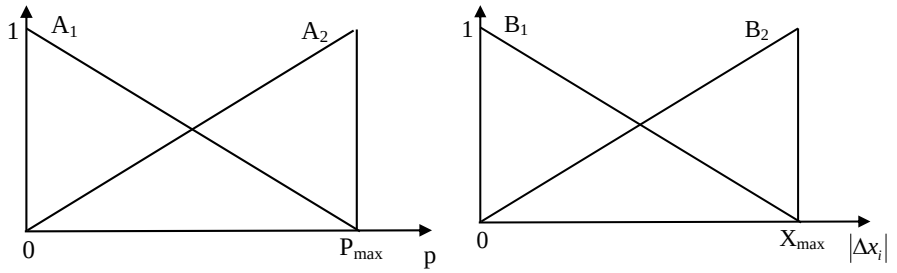

Рис. 1. Пример функций принадлежности термов: A_1 (*низкая*), A_2 (*высокая*); B_1 (*малая*), B_2 (*большая*)

При необходимости количество лингвистических термов, являющихся значениями лингвистических переменных можно увеличить до 3–4. При этом количество правил также соответственно увеличится.

Операционная цена товара определяется агентом-диспетчером на основе дефузифицированных значений потреблений и спроса каждого агента, которые передаются диспетчеру на каждом раунде работы системы. На основе значения текущей цены и спроса (предложения) как агент-потребитель, так и агент-поставщик вычисляют агентскую функцию действия. Поверхность отклика такой функции для агента с двумя переменными показана на рис. 2.

Однако, для задания агентов которые могут оперировать со средой, лишь частично поддающейся наблюдению, традиционных средств нечеткого вывода недостаточно. С этой целью в [9] вводится класс агентов, основанных на модели. Внутри агента хранится представление о той части информации, что находится вне границ его “обзора”. Поэтому для спецификации таких агентов целесообразно использовать моделе-ориентированный подход.

Неточность и неопределенность при спецификация окружающей среды задаются нечеткими графами, а преобразований модели с учетом внутреннего состояния агента - системами трансформаций нечетких графов (СТНГ) [6]. Они являются обобщением последовательных систем трансформаций графов (графовых грамматик) и учитывают основные виды нечеткости, которые возникают как при построении базовых категорий нечетких объектов, так и при описании трансформаций нечетких графов, порождаемых нечеткими множествами.

Формальные модели СТНГ используется при проектировании нечетких агентов с двумя целями: 1) как обобщение алгоритмов нечеткого логического вывода, в частности, таких, как системы Мамдани и Ларсена над нечеткими графами; 2) с целью спецификации возможных преобразований между моделями, задающими разные уровни абстракции нечетких агентов в подходе MDA (так называемая “вертикальные” трансформации или преобразования “уточнения”).

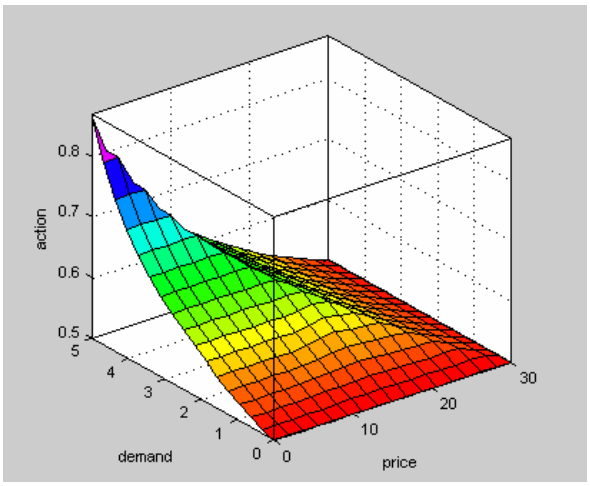

Рис. 2. Пример поверхности отклика агента-потребителя

Каждая продукция СТНГ строится на основе нечетких графов, которые представляют собой антецедент и граф-результат продукции. Такая продукция, может быть записана как указано далее.

ЕСЛИ нечеткий граф-антецедент ТО нечеткий граф-результат.

Возможность указанного представления продукций СТНГ основано на том факте, что нечеткий граф-антецедент и нечеткий граф-результат (консеквент) могут быть записаны в виде

$$\tilde{N}_1 \,\&\, \tilde{N}_2 \ldots \tilde{N}_N \,\&\, \tilde{E}_1 \,\&\, \tilde{E}_2 \,\&\ldots \tilde{E}_M ,$$

где $\{ N_1, N_2 ,\ldots, N_N\}$ – множество вершин нечеткого графа, $\{ E_1, E_2 ,\ldots, E_M\}$ – множество ребер нечеткого графа, а $\tilde{e}$ – нечеткое множество соответствующей вершины или ребра нечеткого графа. Обозначим принадлежность вершины и ребра графу-антецеденту и графу-консеквента соответственно верхним индексом *A* и *C*.

Таким образом, простая продукция, задающая соответствующее преобразование, может быть записана в виде

$$\text{ЕСЛИ } \tilde{N}_1'^A = \tilde{N}_1^A \,\&\ldots\&\, \tilde{N}_N'^A = \tilde{N}_N^A \,\&\, \tilde{E}_1'^A = \tilde{E}_1^A \,\&\ldots\&\, \tilde{E}_M'^A = \tilde{E}_M^A$$
$$\text{ТО } \tilde{N}_1'^C = \tilde{N}_1^C \,\&\ldots\&\, \tilde{N}_K'^C = \tilde{N}_K^C \,\&\, \tilde{E}_1'^C = \tilde{E}_1^C \,\&\ldots\&\, \tilde{E}_L'^C = \tilde{E}_L^C$$

где $\tilde{N}_i^A$, $\tilde{E}_j^A$, $\tilde{N}_i'^A$, $\tilde{E}_j'^A$, $\tilde{N}_k^C$, $\tilde{E}_l^C$, $\tilde{N}_k'^C$, $\tilde{E}_l'^C$ – дискретные нечеткие множества или нечеткие числа, причем $\tilde{N}_i^A$, $\tilde{E}_j^A$, $\tilde{N}_k^C$, $\tilde{E}_l^C$ – указанные в продукции нечеткие множества, $\tilde{N}_i'^A$, $\tilde{E}_j'^A$ – значения истинности вершин и ребер нечеткого графа к которому применяется продукция, обычно отличающееся от значений указанных в продукции; $\tilde{N}_k'^C$, $\tilde{E}_l'^C$ – исправленные значения истинности нечетких множеств графа получаемого в результате применения продукции.

Пусть *P* – значение истинности антецедента продукции, *NA* – множество вершин, а *EA* – множество ребер графа-антецедента. Тогда

$$P = \min\begin{cases}\min_{n\in NA}(\max(\min_{\forall x}(\mu'(n,x),\mu'(n,x)))),\\ \min_{e\in EA}(\max(\min_{\forall x}(\mu'(e,x),\mu'(n,x)))).\end{cases}$$

Реализовано три типа вывода над нечеткими графами: монотонный, при котором последовательные значения истинности вершин и ребер могут только возрастать; немонотонный, при котором последовательные значения истинности могут как расти, так и уменьшаться; и нисходящий монотонный, при котором последовательные значения истинности только уменьшаться. Адекватность применения определенного типа вывода зависит от задачи и может отличаться для различных продукций, соответствующим определенным этапам решения.

При монотонном выводе существующие значения истинности вершин и ребер графа, не могут быть уменьшены при наличии дополнительных свидетельств. Пусть *NC* – множество вершин, а *EC* – множество ребер графа-консеквента. Формула монотонного вывода для значений истинности вершин и ребер, которые добавляются или остаются в соответствии с указанной продукцией:

$$\forall o \in NC \cup EC: \;\; \mu'(o,x) = S(P, \mu(o,x)) ,$$

где функция *S* – так называемая *S*-норма, например $S(P, \mu(o,x)) = \max(P, \mu(o,x))$.

При немонотонном выводе, мы допускаем, что новые результаты, обеспечиваемые запуском указанной продукцией, надежнее, чем любые существующие свидетельства:

$$\forall o \in NC \cup EC: \ \mu'(o,x) = P .$$

Нисходящий монотонный вывод полезен, когда значения истинности $\mu'(o,x)$ представляет верхний предел возможного значения:

$$\forall o \in NC \cup EC: \ \mu'(o,x) = T(P, \mu(o,x)),$$

где функция T – любая T-норма, например, $T(P,\mu(o,x)) = \max(P,\mu(o,x))$, что соответствует оператору вывода Мамдани.

Структура нечеткого агента, основанного на мотивации приведена на рис. 3. В данной реализации используются системы СТНГ для того, чтобы непосредственно отображать стимулы X на ответы R. Выбор этого способа был основан на простоте наличия единственной системы нечетких правил, обеспечивающей функцию отображения β, так как этот выбор не требует механизма координации. Отсутствие координатора поведения внутри самого агента упрощает его реализацию и делает более очевидным эффект влияния нечетких значений побуждения на результирующее поведение агента.

В случае четких значений входных стимулов $x = \{x_1,\ldots x_n\}$ фуззификатор приводит их к соответствующим нечетким значениям. Блок нечеткого вывода осуществляет сопоставление нечетких правил СТНГ не только с нечеткими стимулами $\tilde{x} = \{\tilde{x}_1,\ldots \tilde{x}_n\}$ но и с нечеткими значениями мотивации $\tilde{m} = \{\tilde{m}_1,\ldots \tilde{m}_N\}$. Для этого задана проекция нечетких правил нечеткого графа предметной области (ПО) нечеткое множество побуждений, то есть, отображение, определяющее какие из вершина графа ПО могут рассматриваться в качестве мотивов (побуждений). Правила СТНГ, в левой части которых находятся нечеткие значения мотивов рассматриваются как строго мотивированные. При использовании нисходящего нечеткого вывода высота функции принадлежности мотивации в правой части правила может только уменьшаться, что приведет к остановке выполнения СТНГ через несколько итерации ввиду отсутствия необходимых побуждений (мера нечеткого совпадение меньше необходимого порога). Дефуззификатор j-го агента отображает множество вершин нечеткого графа ПО, соответствующих нечетким значениям поведения во множество одновременных действий $\beta_j = \{\beta_{j1},\ldots \beta_{jl}\}$.

На рис. 4 показана архитектура нечеткой мультиагентной системы, состоящей из N мотивированных агентов. Все агенты получают одинаковые стимулы $\tilde{x} = \{\tilde{x}_1,\ldots \tilde{x}_m\}$. Каждый агент осуществляет применение правил СТНГ и вырабатывает решение относительно приемлемого поведения $\beta_j = \{\beta_{j1},\ldots \beta_{jl}\}$. Выработка окончательного решения осуществляется координатором, задачей которого также является также настройка нечетких правил. С этой целью, модуль оценки предложений агентов на основе $\beta_1, \beta_2, \ldots, \beta_N$, набора соответствующих этим предложениям входных символов и набора мотиваций осуществляет настойку правил СТНГ для каждого агента и передает полученный результат модулю вычисления результата. В результате сопоставления с использованием индексов нечеткого ранжирования вырабатывается окончательное решение мультиагентной системы, означающее “наилучшее” предложенное поведение $\beta_i = \{\beta_{i1},\ldots \beta_{il}\}$.

В нашей реализации координатора мультиагентной системы начальные установки побуждения используются для того, чтобы на основе системы нечетких правил Такаги-Сугено определить нечеткую пригодность поведения агента в различных средах (рис. 5). Множество побуждений M нечеткого агента для закупки (продажи) товаров и услуг включает: отсутствие у агента необходимого количества товаров (услуг) (m_1), избыток товаров (m_2), большой срок хранения товаров или пользования услугой (m_3) и возможность закупки/продажи у минимального количества поставщиков/потребителей (m_4). Эти побуждения используются как входные установки (нечеткие числа) до запуска каждого эксперимента.

Для возможности такой настройки пользователь выбирает значения побуждений (т.е. m_1, ...,m_N) которые должны быть использованы в нечетком вычислении полезности (т. е. побуждения, которые помогают выбрать лучшего агента в популяции) одну из допустимых сред обучения агента.

Модуль оценивания предложений агентов осуществляет следующие шаги:

– на основе генетического алгоритм генерирует популяцию различных мотивированных агентов;

– каждый агент популяции, в свою очередь, вырабатывает и выполняет последовательность набор действий β_j в среде;

– на основе предложенного поведения каждого агента координатор производит набор N значений полезности;

– результирующая нечеткая полезность F вычисляется на основе побуждений m_1, ...,m_N и отдельных значений полезности $f_1,\ldots f_N$.

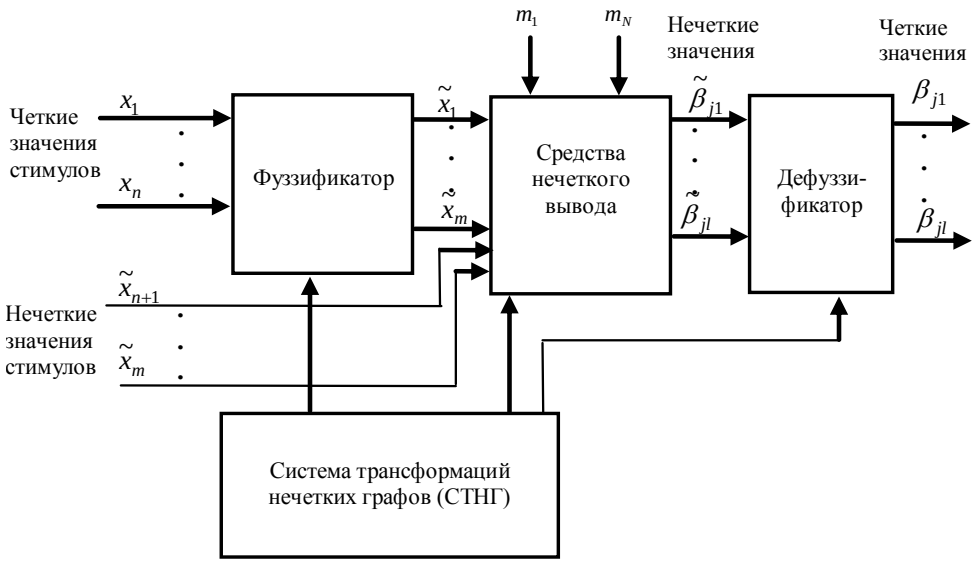

Рис. 3. Структура нечеткого мотивированного агента

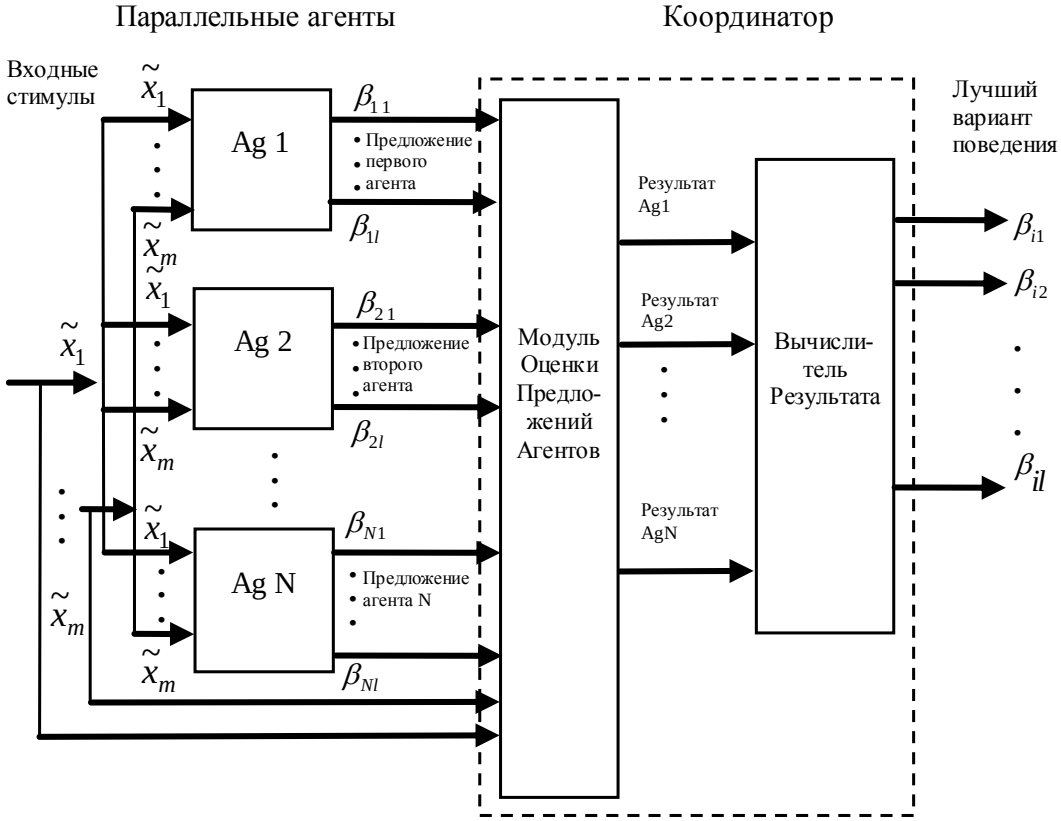

Рис. 4. Структура нечеткой мультиагентной системы с координатором

Оценивание заканчивается, когда достигнуто число максимальных повторений генетического алгоритма. Лучший агент в популяции сохраняется как заключительный вариант агента. Для определения нечеткого значения полезности для специфического поведения, используется не требующая дефуззификации система Такаги-Сугено. Результатом каждого правил являются четкие значения, суммируемые как взвешенная средняя величина [14]. При определении значения полезности для каждого из четырех побуждений торгового агента использованы треугольные функции принадлежности, показанные на рис. 6. Таким образом, использованы четыре нечетких переменных с пятью функциями принадлежности каждая (5^4 = 625 различных нечетких правила). Соответствующий алгоритм расчета значений нечеткой полезности показан на рис. 7.

Для торгового агента определены следующие переменные и соответствующие им критерии: надлежащее окончание действий в состоянии, при котором значение ресурсов (имеющихся товаров или услуг).

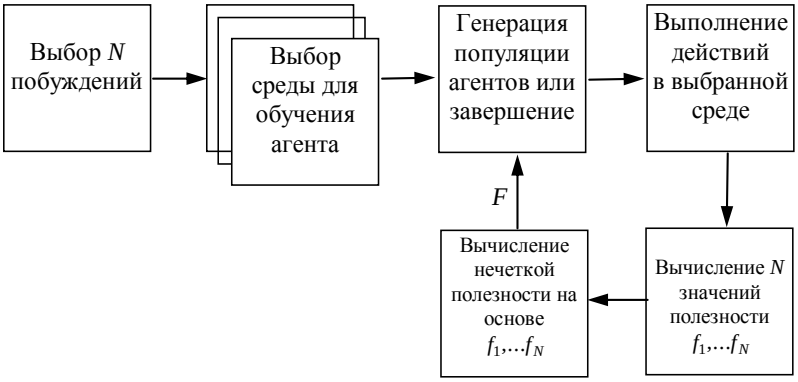

Рис. 5. Схема настройки нечетких правил на основе побуждений и функции полезности

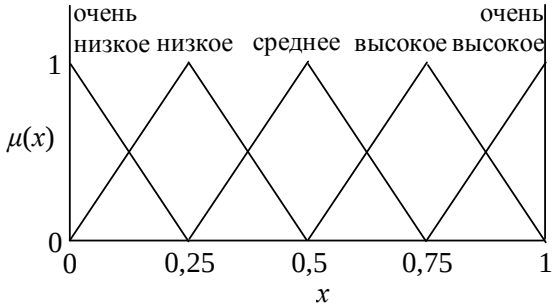

Рис. 6. Нечеткие функции принадлежности

Входные значения: N : количество мотиваций;
M : количество функций принадлежности каждой из мотиваций;
X[N] : массив заданных мотиваций;
Y[N] : массив значений функций полезности;
C[N] : массив коэффициентов;
μ[N][M] : матрица функций принадлежности для мотиваций;
Переменные:
w[n] : оцениваемый вес каждого нечеткого правила; f[n] : определяемая полезность;
n, m0, m1, . . . , mN : целые значения;
Выходные значения:
F : нечеткое значение полезности;
НАЧАЛО
n := 1;
ДЛЯ $m_1, m_2, \ldots, m_N$ ОТ 1 ШАГ 1 ДО *M*
  НАЧАЛО
    w[n] := min{μ[1][m1], μ[2][m2], . . . , μ[N][mN]};
    f[n] := $\sum_{i=1}^{N} X[i]Y[i]C[m_i]$;
    n := n + 1;
  КОНЕЦ;
F := $(\sum_{i=1}^{M^N} w[i]f[i])/(\sum_{i=1}^{M^N} w[i])$;
КОНЕЦ

Рис. 7. Алгоритм вычисления нечеткого значения полезности

не меньше установленного для агента уровня (f_1), не больше установленного для агента уровня (f_2), необходимость предпринимать дальнейшие операции из-за ограничения срока хранения или расходования товаров или ресурсов (f_3), количество поставщиков/потребителей, с которыми были проведены операции (f_4). Значения для этих критериев нормализованы (содержат нечеткие числа) и вычислены после того, как агент завершает действия.

В процессе обучения, управляемая среда выбрана, и генетический алгоритм случайным образом генерирует начальную популяцию агентов. После этого, каждый агент выполняет свое задание (покупка и продажа) и вырабатывается набор значений полезности, соответствующих выполненному заданию. В результате на основе значений побуждения (мотивации) и полученных значений полезности по отдельным критериям, координатор вычисляет значение нечеткой полезности.

Процедура согласования поведения нечетких агентов в мультиагентных системах

В большинстве распределенных мультиагентных систем для выработки рационального поведения одних нечетких моделей недостаточно. Любой агент должен вести себя так, чтобы максимизировать ожидаемое значение функции полезности [9].

Процедура согласования поведения нечетких агентов основана на обмене информацией между агентами относительно нечетких атрибутов для объектов поставки/потребления. Она включает в себя два основных этапа: 1. Анализ и оценка имеющихся предложений средствами нечеткой логики. 2. Генерация дальнейших запросов с учетом предпочтений обеих сторон.

Агент-потребитель анализирует предложение агента-поставщика на основе его собственной нечеткой базы знаний. Первый шаг - загрузить правила базы знаний и получить нечеткие требования пользователя.

Пусть имеется n согласовываемых атрибутов (например, цена, время поставки, количество и т.д.), а согласовываемый атрибут обозначим Ni, $i = 1, 2, \ldots, n$. Каждый такой атрибут может принимать возможное значение j (например, дешевый, дорогой), причем $j = 1, 2, \ldots, m$. Трапециевидная нечеткая функция принадлежности значения j согласовываемого атрибута i представляется на основе четырех параметров a_{Nij}, b_{Nij}, c_{Nij}, d_{Nij} следующим образом:

$$F_{Nij}(x) = \begin{cases} 1, \text{ если } b_{Nij} \le x \le c_{Nij} \\ (x - a_{Nij})/(b_{Nij} - a_{Nij}), \text{ если } a_{Nij} \le x \le b_{Nij}, \\ (d_{Nij} - x)/(d_{Nij} - c_{Nij}), \text{ если } c_{Nij} \le x \le d_{Nij}, \\ 0, \text{ если } x \le a_{Nij} \text{ или } x \ge d_{Nij}. \end{cases}$$

В некоторых случаях существует некоторые лингвистические значения, такие как, например, цена, которые монотонно увеличиваются или уменьшаются. Например, если клиент, устанавливает в запросе атрибут “как можно дешевле”, то агент потребует от пользователя, уточнить предполагаемые параметры a_{Nij} и b_{Nij} этого запроса:

$$F_{Nij}(x) = \begin{cases} 1, \text{ если } x \le a_{Nij} \\ (b_{Nij} - x)/(b_{Nij} - a_{Nij}), \text{ если } a_{Nij} \le x \le b_{Nij}. \\ 0, \text{ если } x \ge b_{Nij}. \end{cases}$$

Агент генерирует экранную форму для того, чтобы выяснить требования пользователя. Запрос включает требования относительно согласовываемых атрибутов Ni. Если пользователь требует их согласования с лингвистическим значением j, чтобы проанализировать предложения агентов-поставщиков применяется нечеткая функция принадлежности $FNij(x)$. Вес требования (W_{Nj}) указывает оценку пользователем значимости каждого атрибута в запросе:

$$\sum W_{Nj} = 1, \text{ где } j = 1, 2, \ldots, n \text{ и } W_{Nj} \in \{0,1\}.$$

Начальное значение $D(t)$ представляет приемлемый уровень решения. Агент регулирует это значение динамически в соответствии с действиями агента-поставщика. t =1, 2, .., T_{max} указывает на t-й раунд согласования. Для того, чтобы ускорить выполнение согласования, агент регулирует значение приемлемого уровня для различных атрибутов. Приемлемое значение согласования атрибута $N_i - D_{Ni}(t)$.

Принятие предложений агентом ограничивается теми случаями, когда все рассчитанные нечеткие индикаторы выше опорного значения B для каждого атрибута.

Предположим, что предложение относительно согласовываемого атрибута Ni – $ONi(t)$. Поэтому, согласно требований пользователя, функция принадлежности $FNij(x)$ используется для расчета нечеткой тождественности требования пользователя предложению. Уравнение для вычисления нечеткой тождественности i-го атрибута определяется на универсальном множестве X следующим образом:

$$M_i = \frac{1}{Card(X)} \int_X [\min((F_{Nij}(x) \Rightarrow O_{Ni}(t)), (O_{Ni}(t) \Rightarrow F_{Nij}(x)))] dx .$$

Обобщенный нечеткий индикатор M адекватности конкретного предложения будет вычислен как взвешенное среднее значений Mi:

$$M = \frac{\sum_i M_i * W_{Ni}}{\sum_i W_{Ni}}$$

Если *M* больше, чем приемлемый уровень *D,* и все нечеткие индикаторы по каждому индивидуальному атрибуту выше, чем базовое значение, то агент принимает предложение немедленно.

Вариант 1: Принять предложение. Агент примет предложение $O(t)$ если $\forall M_i \geq B_i \wedge M \geq D(t)$.

Вариант 2: Проводить согласование далее. Агент будет далее вести переговоры с поставщиком, если $M < D(t)$.

Чтобы оценить предложение $O(t)$, агент вычисляет значение "преимущества" $GNi(O(t))$ для индивидуального атрибута согласования *Ni* по формуле:

$$G_{Ni}(O(t)) = F_{Nij}(x) - D_{Nij}(t).$$

В случае $GNi(O(t)) > 0$ предложение согласовываемого атрибута *Ni* выгодно агенту-потребителю. Равенство $GNi(O(t)) = 0$ показывает, что предложение согласовываемого атрибута *Ni* точно соответствует требованию потребителя в момент времени *t* (поскольку приемлемый уровень $DNi(t)$ изменяется со временем). Ситуация $GNi(O(t)) < 0$ показывает, что предложение атрибута *Ni* не удовлетворяет требованиям пользователя.

Поскольку процедура согласования включает n нечетких значений, то полное значение преимущества или потерь определяются как взвешенное среднее преимуществ (потерь) по каждой проблеме согласования:

$$G(O(t)) = \sum_i G_{Ni}(O(t)) * W_{Nij},$$

где $i = 1, 2, \ldots, n$. Это уравнение показывает, что потери по некоторым атрибутам могут быть компенсированы получением преимуществ по другим атрибутам. Таким образом, задача опорного значения - предотвратить чрезмерную потерю при специфическом результате согласования.

Задача вычисления нормы уступки следующая: (1) чтобы оценить соотношение согласования, которое будет объяснено далее; (3) чтобы планировать стратегию. Норма средней уступки поставщика по проблеме согласования *Ni* вычисляется следующим образом:

$$C_{Ni} = \frac{O_{Ni}(1) - O_{Ni}(t)}{O_{Ni}(1)}$$

Это уравнение показывает, как вычислить норму уступки по атрибуту ***Ni*** на *t*-th раунде согласования. Однако, цель вычисления этого значения - выяснить, насколько сильно стороны стремятся договориться относительно значения данного атрибута. Поэтому, полная норма уступки по всем атрибутам должна быть нормализована к 1. Другими словами, норма уступки атрибута *Ni* на *t*-м раунде согласования определяется как:

$$C_{RNi}(t) = \frac{C_{Ni}(t)}{\sum_j C_{Nj}(t)}.$$

где $i = 1, 2, \ldots, n$.

Цель вычисления коэффициента согласия – оценить возможность урегулировать запрос по каждому согласовываемому атрибуту для того, чтобы оценить потенциал для дальнейших переговоров между агентами. Он включает: 1) норму уступки $C(t)$ по атрибуту *Ni*; 2) вес атрибута *Ni*:

$$K_{Ni}(t) = \sqrt{C_{RNi}(t)^2 + W_{Ni}{}^2}.$$

Вычисление чистого преимущества (дохода) агентов осуществляется как

$$G_P(O(t)) = \sum_i G_{Ni}(O(t)), \quad \forall G_{Ni}(O(t)) > 0.$$

Этот шаг состоит в нахождении возможного объема уступки. Если агент обнаруживает, что чистое преимущество значительное по некоторым атрибутам, он может уступить большее значение по другим атрибутам в течение процедуры согласования. Изменение потери атрибута в соответствии с его весом, и вычисление соотношения преимущества задается как

$$P_{Ni}(O(t)) = W_{Ni} \frac{G_P(O(t))}{\sum_i W_{Ni}}, \quad \forall G_{Ni}(O(t)) < 0.$$

Поскольку больший вес означает, что результат более важен для пользователя, агент выделяет большую компенсацию для этого атрибута используя значения преимуществ по другим атрибутам (рис. 8). Как только потери по отдельным атрибутам компенсированы, агент пытается уменьшить значение согласования по каждому атрибуту, чтобы достичь консенсуса между поставщиком и потребителем:

$$D_{Ni}(t+1) = K_{Ni}(t) * \max\left[O_{Ni}(t),\ D(t) - \frac{P_{Ni}(O(t))}{W_{Ni}},\ B \right].$$

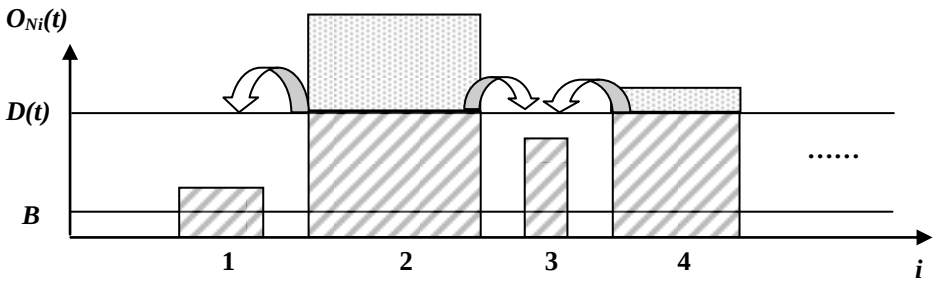

Рис. 8. Компенсация значений атрибута агента

Общее значение согласования $D(t+1)$ на следующем раунде как средневзвешенное значений по каждому атрибуту:

$$D(t+1) = \sum_j W_{NJ} * D_{NJ}(t+1), \text{ где } j = 1, 2, \ldots, n.$$

Функция полезности взаимодействующих нечетких агентов основана на значениях получаемого преимущества $G(O(t))$ и коэффициентах согласия по всем раундам согласования:

$$U_{Ni}(t) = \sum_t K(O_{Ni}(t)) * G_{Ni}(O(t)).$$

Выводы

Таким образом, нами предложены нечеткие архитектурные модели мультиагентных систем, предназначенных для решения задач в распределенной среде. Такие гибридные модели обладают свойствами как реактивных, так и мотивированных делиберативных агентов, что позволяет использовать преимущества обеих подходов. Возможность применения таких моделей иллюстрируется, например, на задаче распределенного управления ресурсами. Предложены средства спецификации поведения нечетких агентов на основе трансформаций нечетких графов и средств адаптации нечетких правил координатором. Формализована процедура согласования поведения нечетких агентов в мультиагентных системах относительно нечетких атрибутов, задающих свойства объектов. Предложена функция полезности, позволяющая получить оптимальное поведение нечетких агентов в процессе многоатрибутного итеративного согласования в распределенной среде.

Уместно отметить, что в качестве платформы реализации данной нечеткой мультиагентной системы используется кластерная система СКИТ-3 [15], в которой агенты взаимодействуют на основе асинхронного обмена сообщениями средствами MPI. Вышеизложенные теоретические результаты в настоящее время применяются для разработки архитектуры нечетких агентов на основе использования MDA (Model Driven Architecture, Архитектуры, управляемой моделями).

1. *Wooldridge M.J.* An Introduction to Multiagent Systems. – Cambridge: MIT Press, 2002. – 366 p.
2. *Основы* инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслов // 2-е изд. – Киев: Академпериодика, 2007. – 672 с.
3. І*нсерційне* програмування / Летичевський О.А., Капітонова Ю.В., Волков В.А., Вишемирський В.В., Летичевський О.О. // Кибернетика и системный анализ. - 2003. - № 1. - С. 19-32.
4. *Летичевський О.А., Капітонова Ю.В., Летичевський О.О., Котляров В.П., Нікітченко М.С., Волков В.А., Вейгерт Т.* Інсерційне моделювання в проектуванні розподілених систем // Проблеми програмування 2008 (4). pp. 13-38.
5. *Заде Л.А.* Роль мягких вычислений и нечеткой логики в понимании, конструировании и развитии информационных/интеллектуальных систем. - Новости Искусственного Интеллекта, №2-3, 2001, с. 7 - 11.
6. *Парасюк І.М., Єршов С.В.* Методи аналізу програмних архітектур, представлених нечіткими графовими моделями // Проблемы программирования. – 2006. – № 1–2. – С. 101–110.
7. *Ершов С.В.* Нечеткие графы функциональных зависимостей как основа метамоделирования программных систем // Компьютерная математика. – 2005. – № 3. – С.139–149.
8. *Ершов С.В.* К проблеме формализации объектно-ориентированных методов разработки программного обеспечения на основе нечеткой логики // Там же. – 2003. – № 2. – С. 62–77.
9. *Рассел С., Норвиг П.* Искусственный интеллект. Современный подход. : Вильямс, 2007. – 1408 с.
10. *Bussmann S., Jennings N., Jennings N.R., Wooldridge M.J.* Multiagent systems for manufacturing control: a design methodology 2004 - 288 p.
11. *Luck M.M., Ashri R., D'Inverno M.* Agent-based software development, 2004 - 208 p.
12. *Anumba C.J., Ugwu O. O., Ren Z.* Agents and multi-agent systems in construction, 2005 - 329 p.
13. *Plekhanova V.* Intelligent agent software engineering. - 2003 - 240 p.
14. *Рутковская Д., Пилиньский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия – Телеком, 2004. – 452 с.
15. *Суперкомпьютеры* ИК НАН Украины. – http:// /icybcluster.org.ua.