

Гибрид агент-ориентированной модели оценки знаний участниками дистанционного обучения

Виктор Борисович Артеменко
доцент, к.э.н., профессор кафедры информационных систем в менеджменте,
Львовская коммерческая академия,
ул. Туган-Барановского, 10, г. Львов, 79005, (032)2958147
artem@lac.lviv.ua

Аннотация

В данной статье рассматриваются методологические подходы к построению гибридной агент-ориентированной модели (АОМ) оценивания знаний в сфере дистанционного обучения (ДО) с использованием искусственных нейронных сетей. Речь идет о разработке теоретической модели, которая направлена на поддержку компьютерных экспериментов оценивания тенденций производства и распространения знаний участниками ДО на примере одного из ВУЗов. При разработке искусственных нейронных сетей нами применялся переход к таким унифицированным шкалам измерения переменных, чтобы нулевое значение преобразованной переменной модели соответствовало наименьшему качеству ее характеристики, а единичное – наибольшему. Исследуется эффективность использования программного обеспечения для поддержки реализации гибрида АОМ: *AnyLogic*, одного из популярных пакетов для агент-ориентированного моделирования, а также пакета *STATISTICA Neural Networks*.

This article discusses the methodological approaches to the construction of a hybrid agent-oriented model (ABM) to estimate the knowledge in the field of distance learning (DL) or e-learning using artificial neural networks. This is a theoretical model, development of which is aimed at supporting computer simulations to assess the trends of production and dissemination of knowledge to the DL as an example of one of the universities. In designing artificial neural networks, we use the transition to such a standardized scales measuring the variables to zero value of the transformed variable model corresponds to the least quality of its characteristics, and identity - the greatest. Investigate the effectiveness of software to support the implementation of hybrid ABM: *AnyLogic*, one of the most popular packages for agent-based modeling, as well as a package *STATISTICA Neural Networks*.

Ключевые слова

дистанционное обучение, гибрид агент-ориентированной модели, программное обеспечение агент-ориентированного моделирования, нейронные сети;
distance learning, hybrid agent-based model, software agent-based modeling, neural networks.

Введение

О знании как особом продукте, который является уникальным и не исчезает после потребления (подобно информации), написано немало, начиная с классической работы Ф. Махлупа [1]. Особенно подчеркивается, что измерять знания не так просто, как измерять количество обычных продуктов [2]. Один подход – это мерить по затратам, как это принято при измерении стоимости продуктов. Естественно, что полные затраты получить чрезвычайно трудно. Другой подход – измерение количества знания по результатам, например, по числу людей потребивших знание. Если знание никем не используется, то логично считать, что его ценность (или полезность) равна нулю. Чем большее количество людей использует знание, тем выше его ценность.

Отметим, что в сети Интернет измерение количества знаний производится по результатам, для оценки которых наиболее распространенными являются индексы цитирования, количество кликов или копий и другие [3]. С помощью подобных способов измерения может обеспечиваться оценка знаний конкретными участниками и в сфере дистанционного обучения (ДО).

В статье [4] рассмотрены методологические и методические подходы к моделированию коммуникативных взаимодействий трех типов агентов e-learning (электронного обучения): авторов, тьюторов и студентов дистанционных курсов (ДК). Средой их коммуникативного взаимодействия является виртуальный центр одного из высших учебных заведений [5]. Создание агент-ориентированной модели направлено на поддержку компьютерных экспериментов оценивания тенденций производства и распространения знаний участниками ДО.

Агент-ориентированные модели (далее АОМ) или Agent-Based Model (ABM) – это новые инструментальные средства для извлечения знаний в различных сферах деятельности. При разработки АОМ следует придерживаться следующих основных требований [6, с.49]: автономия агентов микроуровня; ограниченная рациональность их поведения (то есть максимизация функции полезности, учитывая широкий круг возможностей, исключена); функционирование внутри определенной «среды обитания». Конечной целью разработки АОМ является попытка получить представление о некотором наборе правил поведения конкретных агентов, которые представляют собой искусственное общество. В энциклопедии «Википедия» дается для этого понятия следующее определение: «*Искусственное общество* – это агент-ориентированная модель, разработанная для компьютерных симуляций в социальном анализе».

Среди программного обеспечения для построения АОМ [7] можно выделить такие наиболее популярные пакеты – SWARM и AnyLogic. Первый пакет является *open source* коллекцией программных библиотек, созданных в Институте Санта Фе и доступных на его сайте [8]. Некоторые из библиотек написаны с помощью языка скриптов, обеспечивающих использование графических средств (графика, окна и др.). AnyLogic – это коммерческий пакет, его специальная библиотека классов *AnyLogic agent based library* позволяет задавать необходимую функциональность у агентов модели [9, 10].

Цель данной статьи – рассмотреть методологические подходы к построению гибридной АОМ оценки знаний в сфере дистанционного обучения с использованием нейронных сетей, одного из направлений искусственного интеллекта. Создание этого гибрида модели направлено на поддержку проведения компьютерных экспериментов с целью оценивания тенденций производства и распространения знаний участниками ДО. Представлены результаты исследований программного обеспечения реализации гибридной АОМ: *AnyLogic* и *STATISTICA Neural Networks*.

Методологические подходы к построению гибридной модели оценки знаний агентами дистанционного обучения

Спецификация задачи разработки гибридной АОМ направлена на построение искусственного общества, в котором взаимодействуют такие основные типы агентов: А1 – авторы ДК, А2 – тьюторы, сопровождающие учебный процесс и А3 – студенты, участники дистанционных курсов. Другими словами, исходя из приведенных выше определений, будем считать искусственным (виртуальным) обществом совокупность агентов микроуровня, включенных в состав АОМ. Агенты искусственного общества ведут себя достаточно автономно: принимают решения, совершают действия и взаимодействуют с другими агентами. Действия агентов являются их ответом на изменения среды и действия других агентов. Ключевой аспект – это взаимодействие между агентами. Для осуществления действия достаточно любого решения агента. Для осуществления же взаимодействия необходимо групповое решение. Групповые решения, как известно, принимаются с учетом определенных правил, которые и необходимо специфицировать в нашем случае.

Основные допущения при разработке модели заключаются в следующем:

- с точки зрения принятия решений, агенты (участники) ДО движутся в двумерном пространстве и имеют конечный горизонт виденья;
- агенты дистанционного обучения взаимодействуют в виртуальной учебной среде по определенным правилам, имеют конечный срок жизни;
- цель автора дистанционного курса (агента 1-ого типа) – произвести как можно больше знаний и передать их тьютору и студентам, цель тьютора (агента 2-ого типа) – распространить знания среди как можно большего числа студентов, а цель студентов (агентов 3-его типа) – употребить как можно больше знаний.

Формализованное описание поведения агентов и результаты компьютерного моделирования в среде *SWARM* представлены в работе [4].

Мы ж хотим исследовать эффективность использования пакета *AnyLogic* для разработки гибридной АОМ оценки знаний участниками ДО на основе нейронных сетей – одного из новых направлений искусственного интеллекта. Обзор научных работ, посвященных этому направлению, позволяет сделать выводы, что результаты нейросетей, обученных на большом количестве наблюдений, больше соответствуют действительности, чем экспертные системы (в которых вычисляются предикаты из базы знаний, полученной путем опроса незначительного количества экспертов) или системы нечеткой логики (используя правила, которые также заложены несколькими людьми).

На рис. 1 представлена общая схема гибридной АОМ участников ДО. Из этого рисунка видно, что в процессе создания гибридной модели мы рассматриваем три искусственных общества, каждое из которых представляет собой совокупность лиц, принимающих решения по выработке и распространению в виртуальной учебной среде *Moodle* электронных знаний. Поддержка решений обеспечивается с помощью искусственных нейронных сетей.

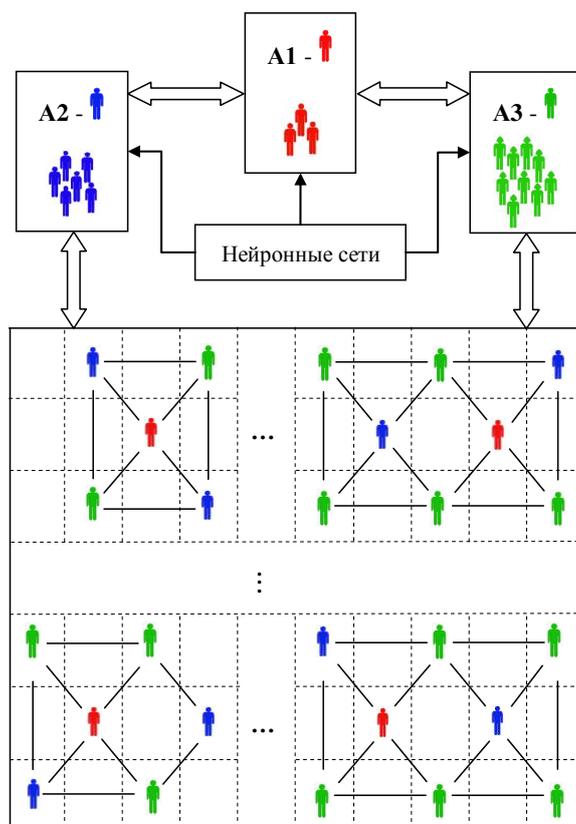


Рис. 1. Общая схема гибрида модели.

Для создания нейронных сетей с последующим их встраиванием в гибридную АОМ, нами использовались данные, которые характеризуют активность извлечения знаний агентами ДО. Через количество записей (кликов), опубликованных в журнале событий системы Moodle, можно проводить годовой мониторинг (на протяжении 12-ти последних месяцев) таких основных действий участников обучения: *просмотр, обновление, добавление, удаление* ресурсов и заданий в дистанционных курсах или на главной странице анализируемого виртуального центра обучения. Следует отметить, что в нашем исследовании применяются четырехмесячные данные, характеризующие 117 000 действий более 500 агентов дистанционного обучения, среди которых около 10% составляют авторы дистанционных курсов, примерно 20% – тьюторы курсов, остальные – студенты.

Все данные, характеризующие клики в журнале событий системы управления дистанционным обучением Moodle, разбиваются на такие непересекающиеся группы данных о кликах (активности): (1) авторов ДК, (2) тьюторов, а также (3) студентов, участников ДК.

На основе этих данных определяется количество кликов агента j -ого типа $\tilde{x}^{(j)}$ ($j=1,2,3$). Каждая группа включает также некоторое условно эталонное значение $\tilde{x}_0^{(j)}$, которое определено внутри заданного диапазона изменения этого показателя (между наименьшим $\tilde{x}_{\min}^{(j)}$ и наибольшим $\tilde{x}_{\max}^{(j)}$ возможным значением) и соответствует лучшему качеству. При этом за условно эталонное значение можно брать экспертные оценки или определенные нормативные показатели.

Предлагается для каждого анализируемого показателя конкретной группы $\tilde{x}^{(j)}$ ($j=1,2,3$) такое преобразование (*), в результате которого область его возможных значений определяется отрезком $[0, 1]$. При этом нулевое значение преобразованного показателя $x^{(j)}$ будет означать наименьшее качество определенной характеристики, а единичное – наибольшее.

Итак, если показатели количества кликов специфицированы под определенные выше группы, то перейти к нормируемым показателям $x^{(j)}$, связанным с входными переменными $\tilde{x}^{(j)}$, можно с помощью следующего преобразования:

$$x^{(j)} = 1 - \frac{|\tilde{x}^{(j)} - \tilde{x}_0^{(j)}|}{\max\{(\tilde{x}_0^{(j)} - \tilde{x}_{\min}^{(j)}), (\tilde{x}_{\max}^{(j)} - \tilde{x}_0^{(j)})\}} \quad (*)$$

Очевидно, что для каждой j -ой группы преобразованная (пронормированная) переменная $x^{(j)}$ может принимать значения от $x^{(j)} = 0$ (соответствует наихудшему качеству) до $x^{(j)} = 1$ (соответствует наилучшему качеству).

На основе выполненного преобразования (*) можно переходить к построению искусственных нейронных сетей. Этот вопрос решается в два этапа:

- 1) выбор типа (архитектуры) сети,
- 2) подбор весов (обучение) сети.

На рис. 2 представлены этапы построения и применения нейронных сетей для последующего их встраивания в АОМ.

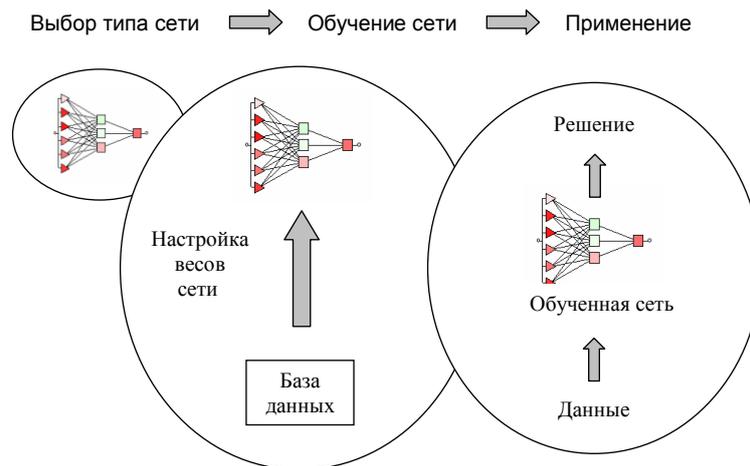


Рис. 2. Этапы построения и применения нейронных сетей.

На первом этапе, для каждого типа (или группы) агентов ДО, необходимо из различных нейросетевых архитектур выбрать эффективную. Наиболее популярные и изученные архитектуры – многослойный персептрон, линейные сети, вероятностная нейросеть, обобщенная регрессионная нейронная сеть и другие.

На втором этапе следует «обучить» выбранную сеть, то есть подобрать такие значения ее весов, чтобы сеть работала нужным образом. Для многих архитектур разработаны специальные алгоритмы обучения, которые позволяют настроить веса сети определенным образом (табл. 1).

Таблица 1

Основные типы нейронных сетей и алгоритмов их обучения

<p><i>Многослойный персептрон (МП).</i> Трехслойный МП (есть 1 скрытый слой) практически моделирует задачи любой сложности. Сети МП можно обучать на основе таких алгоритмов:</p> <ul style="list-style-type: none"> - Метод сопряженных градиентов, - Квази-ньютоновский, - Левенберга-Маркара, - Обратное и Быстрое распространение, - Дельта-дельта с чертой. 	<p><i>Радиальная базисная функция (РБФ).</i> Эти сети имеют входной слой, скрытый слой с радиальными элементами и выходной слой линейных элементов. РБФ относительно быстро учатся на базе линейной оптимизации. Также можно использовать и алгоритмы обучения МП. Однако РБФ превышают размеры многослойного персептрона и поэтому работают медленнее.</p>
<p><i>Линейные сети (ЛС).</i> ЛС имеют только два слоя: входной и выходной, последний имеет линейную пост-синаптическую потенциальную функцию и функцию активации. Линейные сети лучше обучаются по методу псевдообратных. Этот метод оптимизирует последний слой любой сети, если он является линейным.</p>	<p><i>Байесовские сети (ВНС, ОРНС).</i> Существует два типа таких сетей:</p> <ul style="list-style-type: none"> - вероятностная нейронная сеть, используется только в задачах классификации, - обобщенная регрессионная нейронная сеть, применяется только в задачах регрессии. <p>Эти сети обучаются на базе кластерных алгоритмов: К-средних и др.</p>

После многократного предъявления примеров о кликах агентов ДО веса сети стабилизируются, причем сеть дает правильные ответы на почти все примеры из базы данных (показатели количества кликов агентов ДО). В таком случае говорят, что «сеть обучена» или «сеть натренирована». В программных реализациях можно видеть, что в процессе обучения величина ошибки (сумма квадратов ошибок по всем выходам) постепенно уменьшается. Когда величина ошибки достигает приемлемого малого уровня, происходит остановка, а обученную сеть считают натренированной и готовой к применению на новых данных.

Важно отметить, что информация, которую сеть имеет о задаче оценивания тенденций производства и распространения знаний агентами в анализируемой среде,

содержится в наборе примеров о кликах агентов ДО. Поэтому качество обучения сети напрямую зависит от количества примеров в обучающей выборке, а также от того, насколько полно эти примеры описывают данную задачу.

Среди программного обеспечения, разработанного для построения нейронных сетей, мы выбрали пакет *STATISTICA Neural Networks*, который позволяет создавать нейронные сети любых типов архитектуры. В пакете имеется инструмент «Мастер решений», который обеспечивает для начинающих пользователей конструирование набора нейронных сетей с наилучшими характеристиками. Можно применять также «Конструктор сетей», обеспечивающий выбор и обучение нейронных сетей с учетом требований продвинутых пользователей.

Отметим также, что *STATISTICA Neural Networks* имеет отдельный модуль – генератор кода, который добавляет этому пакету возможность создавать эквивалент обученной сети как (некомпилированный) код либо на языке C/C++, либо на Visual Basic. Каждое вычисление и параметр искусственной нейросети открыты и доступны пользователю для просмотра, копирования или изменения.

Фрагменты кода созданной пользователем нейронной сети можно встраивать как функции для дальнейшего использования кода в других приложениях. Огромное преимущество этой возможности в том, что *STATISTICA Neural Networks* не обязана быть установлена на компьютере, где выполняется сгенерированный код.

На рис. 3 представлены примеры нейронных сетей, разработанные средствами *STATISTICA Neural Networks* для гибридной АОМ. Отметим, они имеют одинаковую архитектуру – это многослойные перцептроны. Различия между этими нейросетями состоят в количестве нейронов скрытого слоя и некоторых других параметрах.

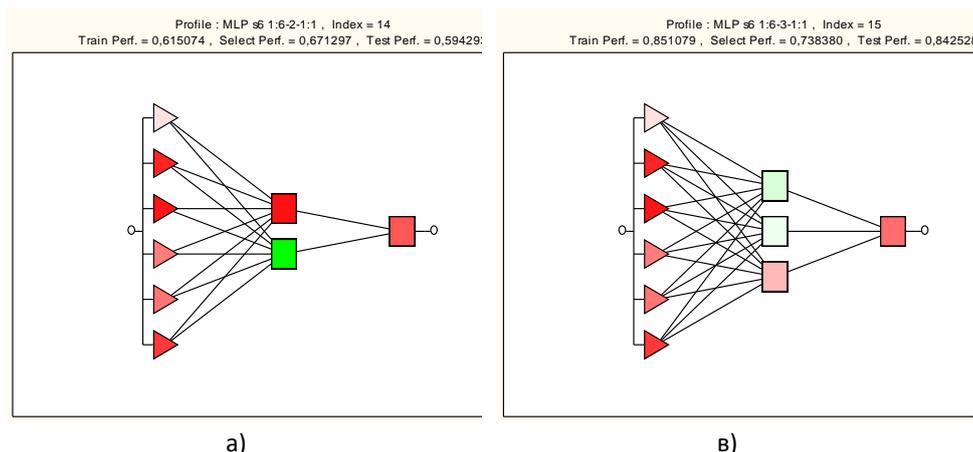


Рис. 3. Архитектуры сетей, предложенные Мастером решений *STATISTICA Neural Networks*.

Числовые значения указанных на рис. 3 параметров нейронных сетей не столь важны, так как акценты в данной работе делаются на методологических подходах к построению гибридной АОМ, и поэтому мы не будем сейчас заострять внимание на технических вопросах реализации нейронных сетей, просто перечислим их основное назначение для создаваемой модели.

- Нейронная сеть №1 (NN#1) – определяет активность агентов 1-ого типа на основе прогнозирования возможного количества кликов и таким образом, оценивает уровень производимого и распространяемого знания авторами ДК.
- Нейронная сеть №2 (NN#2) – определяет активность агентов 2-ого типа на основе прогнозирования возможного количества кликов и таким образом, оценивает уровень распространяемого знания тьюторами ДК.
- Нейронная сеть №3 (NN#3) – определяет активность агентов 3-его типа на основе прогнозирования возможного количества кликов и таким образом, оценивает уровень потребляемого знания студентами, участниками дистанционного обучения.

Следовательно, решения агентов дистанционного обучения как бы принимает соответствующая нейронная сеть, получая на входе переменную, характеризующую возрастание ценности (полезности) знаний для участников искусственного общества определенного типа. При этом количество кликов, измеренное с учетом пороговых значений на перспективу, позволяет в гибридной АОМ (рис. 1) оценивать тенденции производства и распространение знаний агентами на основе результатов активности каждого из участников дистанционного обучения.

Реализация прототипа гибридной АОМ в среде AnyLogic

В настоящий момент в среде *AnyLogic* создан действующий прототип модели, позволяющий фрагментарно реализовывать поведение агентов ДО, представленных на рис. 1, и анализировать их некоторые параметры (конечный срок жизни агентов).

Поведение агентов ДО в нотации *AnyLogic* описывается с помощью активных объектов. Активный объект – сущность, которая инкапсулирует атрибуты объекта, методы и характер поведения объекта как единое целое. Использование активных объектов является естественным средством структуризации модели производства и распространения знаний участниками дистанционного обучения: это искусственное общество состоит из определенного множества параллельно функционирующих и взаимодействующих между собой сущностей. Различные типы этих сущностей и представляют разные активные объекты.

Чтобы создать модель *AnyLogic*, нужно создать классы активных объектов (или использовать объекты библиотек *AnyLogic*). Определение активного объекта задает шаблон, и отдельные объекты, построенные в соответствии с этим шаблоном (экземпляры активного объекта), могут использоваться затем как элементы других активных объектов: *avtorDK* [...], *tutor* [...], *student* [...]. Всегда один класс в модели *AnyLogic* является корневым. Для него в нашем гибриде АОМ порождается один экземпляр с предопределенным именем **Main**, он и запускается системой *AnyLogic* на выполнение.

Каждый активный объект имеет структуру (совокупность включенных в него активных объектов и их связи), а также поведение, определяемое совокупностью переменных, параметров, стейтчартов (диаграмм состояний) и др. Каждый экземпляр активного объекта в работающей модели имеет свое собственное поведение, свои значения параметров. Он функционирует независимо от других объектов модели, взаимодействуя с ними и с внешней средой.

Используемые для описания гибрида модели переменные имеют одинаковую структуру. На рис. 4 изображена схема, объясняющая индексацию переменных.

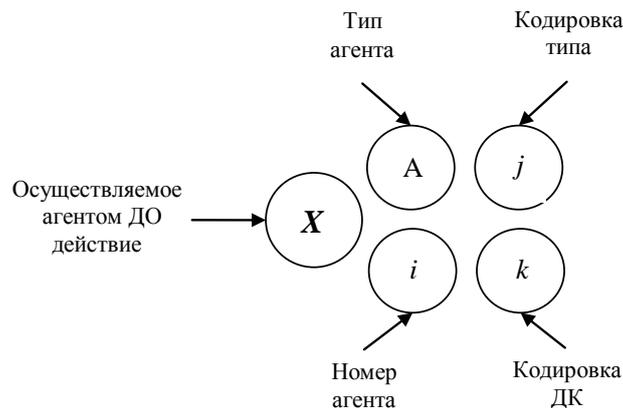


Рис. 4. Обозначение переменных в АОМ.

Символ *X* означает действие (клик), осуществляемое агентом ДО. Примерами такого действия могут быть *P* – просмотр ресурсов и заданий в дистанционном курсе, *O* – обновление ресурсов и заданий в ДК, *D* – добавление ресурсов и заданий в ДК, а

также U – удаление ресурсов и заданий в ДК. Как правило, из контекста понятно, о каком действии идет речь.

Индекс k определяет дистанционный курс, в котором агент осуществляет свои действия. Номер агента дистанционного обучения задается индексом i . Наконец, j – это индекс, означающий тип агента (A1 – автор, A2 – тьютор, A3 – студент).

Следует отметить, что при спецификации поведения агентов анализируемого гибрида АОМ определенные *действия при входе* содержат указание на выполнение кода соответствующей искусственной нейронной сети.

При построении прототипа гибрида АОМ в среде *AnyLogic* использовались такие средства: визуальная разработка (введение состояний и переходов стейтчарта, введения пиктограмм переменных), задание численных значений параметров, а также аналитических записей соотношений переменных и аналитических записей условий наступления событий.

Основной парадигмой, принятой в *AnyLogic* при разработке моделей, является визуальное проектирование – предусматривает построение с помощью графических объектов и пиктограмм иерархий структуры и поведения активных объектов.

Основным средством спецификации поведения объектов в *AnyLogic* являются переменные, таймеры и стейтчарты. Переменные в модели отражают изменяющиеся характеристики конкретного объекта. Таймеры можно взводить на определенный интервал времени и по окончании этого интервала выполнять заданное действие. Стейтчарты позволяют визуально представить поведение объекта во времени под воздействием событий или же условий, они состоят из графического изображения состояний и переходов между ними. Любая сложная логика поведения объектов создаваемого гибрида модели в среде *AnyLogic* может быть выражена с помощью комбинации стейтчартов, переменных, таймеров, программного кода на Java, а также встроенных как функции для дальнейшего использования кодов созданных ранее трех нейронных сетей.

На рис. 5 представлен пример спецификации поведения агентов создаваемого гибрида, отражающий взаимодействие между авторами ДК, тьюторами и студентами, участниками дистанционного обучения.

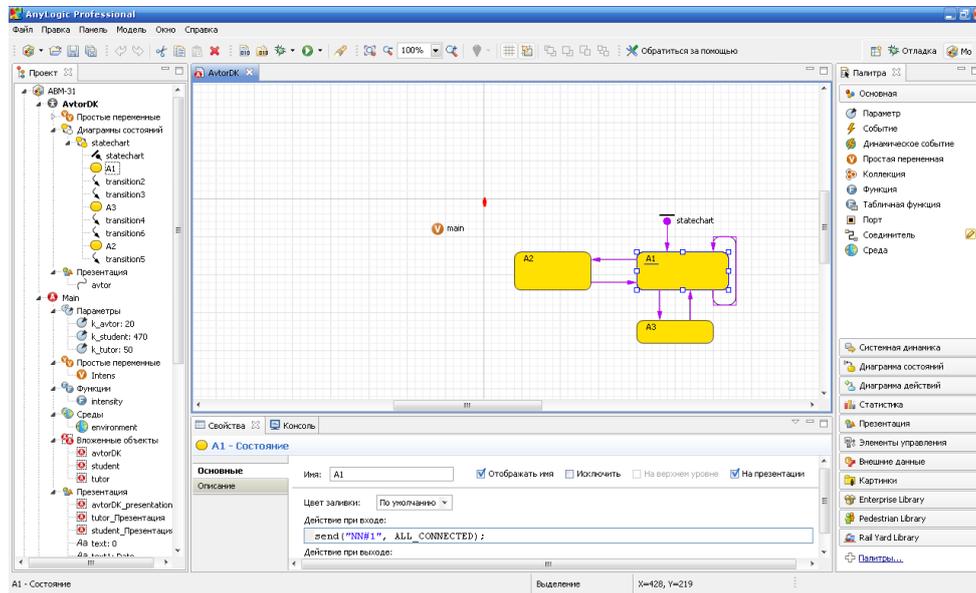


Рис. 5. Пример спецификации объектов АОМ.

Как видно с рис. 5, в строке «Действие при входе» определено взаимодействие агентов с помощью нейронной сети №1 (NN#1), посылающей сообщение о прогнозе количества кликов авторами ДК всем связанным агентам дистанционного обучения и таким образом, оценивающей уровень производимого и распространяемого знания.

Действие при выходе предусматривает такие три случая:

1. Во время встречи двух авторов ДК у обоих происходит прирост знаний пропорционально знанию собеседника.

2. При встрече с тьютором у автора курса прироста знаний не происходит, а происходит прирост показателя, который характеризует передачу знаний. У тьютора происходит прирост знаний пропорционально количеству знаний автора ДК.

3. Во время встречи со студентом у автора происходит прирост (существенно меньше, чем при встрече с тьютором ДК) переданных знаний, а у студента – прирост потребленных знаний.

Поведение тьютора (A2) и студента (A3) описываются в создаваемом гибриде АОМ аналогично с помощью соответственно нейронной сети №2 и нейронной сети №3.

Следует отметить, понятие модельного времени является базовым в агентном моделировании. Модельное время (Time) – условное логическое время, в единицах которого определено поведение всех объектов модели.

В моделях *AnyLogic* модельное время может изменяться либо непрерывно, либо дискретно, переключаясь от момента наступления одного события к моменту наступления следующего события, если в модели присутствуют только дискретные события. Моменты наступления всех планируемых событий в дискретной модели исполнительная система хранит с помощью так называемого календаря событий, выбирая оттуда наиболее раннее событие для выполнения связанных с ним действий. Значение текущего времени в моделях *AnyLogic* может быть получено обращением к функции `getTime ()`.

На рис. 6 представлены страничка настройки компьютерных экспериментов и окно для определения характеристик модельного времени.

Единицу модельного времени в созданном гибриде АОМ мы интерпретируем как день. А модельное время задается с помощью даты начала и конца календаря событий – период обучения во втором семестре текущего учебного года.

Специальным объектом, позволяющим задавать основные свойства, общие для всех типов агентов является среда моделирования. Как правило, в простейшей агент-ориентированной модели присутствует одна среда, заданная на диаграмме корневого объекта эксперимента – там же, куда добавлен и реплицированный объект класса агента, моделирующий агентов. Чтобы добавить агентов в среду, нужно указать имя соответствующего объекта «среда» в свойствах вложенного объекта, моделирующего агентов.

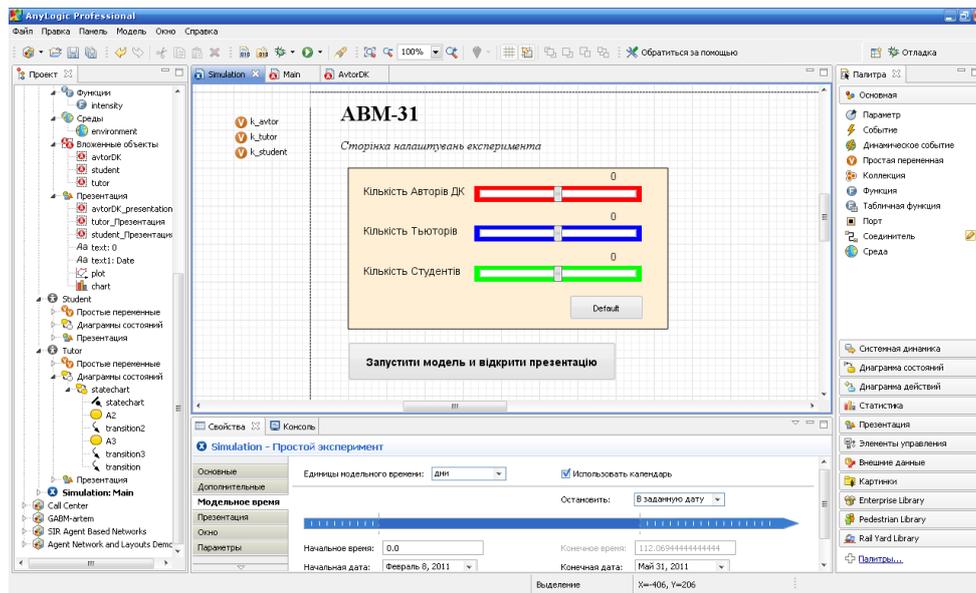


Рис. 6. Страничка настройки экспериментов и модельного времени.

Для того чтобы добавить среду, надо перетащить элемент **Среда** из палитры **Основная** на диаграмму класса активного объекта. Дополнительные ее свойства включают *Тип пространства*, *Тип соседства*, *Тип расположения* и др.

Тип пространства – это группа кнопок, позволяющая задавать следующие типы пространства, в котором будут обитать агенты: *непрерывное*, *дискретное*, *ГИС-пространство* (позволяет размещать агентов в геопространственной среде, заданной с помощью карты геоинформационной системы).

Тип соседства (доступно, если выбрано *дискретное* пространство) – задает модель соседства, которая определяет, какие агенты будут считаться соседями агента в дискретном пространстве. Есть две модели соседства:

- *Мурово*, когда соседями считаются агенты, располагающиеся в 8 соседних ячейках (к северу (NORTH), югу (SOUTH), востоку (EAST), западу (WEST), северо-востоку (NORTHEAST), северо-западу (NORTHWEST), юго-востоку (SOUTHEAST) и юго-западу (SOUTHWEST) от данной).
- *Евклидово*, когда соседями принято считать агентов, располагающихся в 4 соседних ячейках (к северу (NORTH), югу (SOUTH), востоку (EAST), западу (WEST) от данной).

Тип расположения (доступно, если выбрано *дискретное* или *непрерывное* пространство) – задает тип расположения агентов, т.е. как агенты будут изначально расположены в пространстве. Поддерживаются такие типы расположения агентов: *задается пользователем*, *случайное*, *упорядоченное* и др.

На рис. 7 представлены результаты простого эксперимента с разрабатываемым гибридом АОМ оценивания знаний участниками ДО (ABM-31). На презентации можно видеть моделируемых агентов. Каждый агент отображается своей фигуркой, которая имеет определенный цвет. Линиями соединены те агенты, между которыми существуют коммуникативные связи (отметим, что эти связи могут генерироваться случайным образом).

Удобные средства разработки анимационного представления гибрида модели в среде *AnyLogic* позволяют представить функционирование моделируемого процесса производства и распространения знаний в живой форме динамической анимации, что позволяет «увидеть» поведение агентов дистанционного обучения.

Средства анимации позволяют легко создать виртуальный мир (совокупность графических образов анализируемых агентов), управляемый параметрами модели по законам, определенным с помощью уравнений и логики моделируемых объектов.

Временной график, отображающий динамику активности конкретных агентов, участников дистанционного обучения, может быть использован для интерпретации таких характеристик агентов, как конечного срока их жизни в процессе производства и распространения знаний.

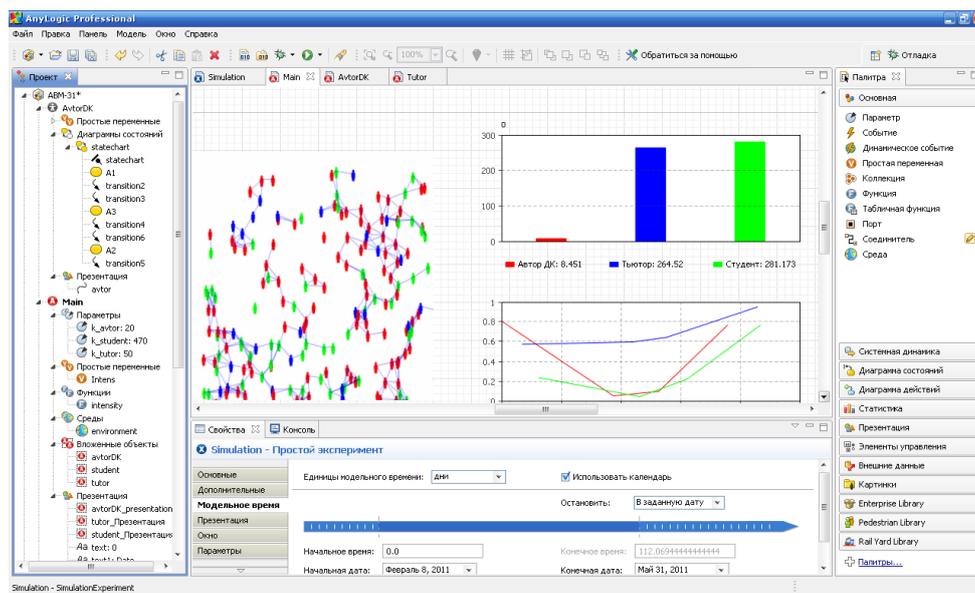


Рис. 7. Результаты компьютерных экспериментов с прототипом создаваемого гибрида АОМ.

Следует отметить, что при построении гибридной модели оценивания знаний агентами дистанционного обучения важным аспектом является формирование среды, специального объекта, позволяющего задавать свойства, общие для группы агентов. Среда гибрида модели должна обеспечивать унифицированный доступ (например, для отправки сообщения какому-то агенту) ко всем агентам, обитающим в среде (без учета принадлежности конкретных агентов к одному и тому же реплицированному (или нереплицированному) вложенному активному объекту). Механизм репликации объектов позволяет создавать любое нужное количество агентов (их число может задаваться параметром) и предоставляет возможность их динамического добавления и удаления.

Программный интерфейс, обеспечивающий доступ к агентам дистанционного обучения, следующий:

- `class AgentCollection extends AbstractCollection<Agent>` – это специальный класс коллекции, позволяющий последовательно проходить по всем агентам и при этом находить агента по его номеру;
- `AgentCollection getAgentCollection()` – возвращает коллекцию всех агентов, населяющих среду;
- `size()` – возвращает количество агентов, населяющих среду;
- `Agent getRandomAgent()` – возвращает случайно выбранного агента;
- `deliverToAll(Object msg)` – немедленно обеспечивает доставку сообщения всем агентам, населяющих среду;
- `deliverToRandom(Object msg)` – немедленно доставляет нужное сообщение случайно выбранному агенту.

Для того, чтобы выполнять с каждым агентом какое-то действие, необходимо будет написать подобный код:

```
for(Agent a : environment.getAgentCollection())  
a.<действие>.
```

Среда гибридной АОМ отвечает и за поддержку пространства. Если в случае непрерывного пространства эта поддержка сводится лишь к расположению агентов в пространстве согласно различным типам их расположения, то при исследовании дискретного пространства среда также отслеживает, какой агент находится в какой ячейке, и предоставляет следующие дополнительные методы:

- `int[...] findRandomEmptyCell()` – возвращает координаты {строка, столбец} псевдослучайно расположенной свободной ячейки (или `null`, если все ячейки заняты);
- `Agent getAgentAtCell(int r, int c)` – возвращает агента, находящегося в заданной ячейке (или `null`, если ячейка пуста).

Заключение

В данной статье речь идет о теоретической АОМ, практическое использование которой остается за рамками нашего рассмотрения. Однако некоторые возможности создаваемой гибридной АОМ оценки знаний участниками дистанционного обучения могут в дальнейшем получить практическое применение.

Результаты исследований предложенных в статье методологических подходов к разработке гибридной АОМ оценки знаний участниками дистанционного обучения позволяют сделать следующие выводы и обобщения с учетом перспектив.

1. Агент-ориентированные модели и искусственные нейронные сети можно рассматривать как эффективные средства для поддержки проведения исследований в сфере дистанционного (электронного) обучения.

2. Результаты компьютерных экспериментов указывают на возможные пути практического применения создаваемого гибрида АОМ для определения:

- временных параметров (для объекта «Расписание занятий») на основе таких характеристик агентов, как конечный срок их жизни) в выполняемом *Open*

Source-проекте «Электронный деканат», направленном на разработку модуля типа «блок» в системе управления дистанционным обучением Moodle [11];

- такой рациональной структуры групп участников е-обучения, при которой количество произведенного и распространенного знания стремится к теоретически максимальному значению;
- потребностей в перепроектировании дистанционных курсов и улучшении их качества на базе разработки и внедрения адаптивных механизмов взаимодействия ключевых агентов е-обучения, для которых важную роль играет использование таких активных компонентов дистанционных курсов, как чаты, вебинары и тематические дискуссии на форумах.

3. В дальнейшем мы намерены продолжить исследовать эффективные пути использования пакета *AnyLogic* для построения рассматриваемого гибрида АОМ с целью оценки производства и распространения знаний участниками дистанционного обучения.

Литература

1. Махлуп Ф. Производство и распространение знаний в США. М.: Прогресс, 1966.
2. Макаров В.Л. Фундаментальная наука и образование: теоретические проблемы интеграции. Гл. 5 // Наука и высокие технологии России на рубеже третьего тысячелетия: социально-экономический аспект развития. М.: Наука, 2001.
3. Паринов С.И. К теории сетевой экономики. Новосибирск: Изд-во ИЭОПП СО РАН, 2002.
4. Артеменко В.Б. Компьютерное моделирование коммуникативных взаимодействий агентов e-Learning // Образовательные Технологии и Общество. 2010. Т.13. №2. – С.345-354. URL: http://ifets.ieee.org/russian/depository/v13_i2/pdf/12r.pdf (дата обращения: 31.03.2011).
5. Веб-центр Львовской коммерческой академии. URL: <http://virt.lac.lviv.ua/> (дата обращения: 31.03.2011).
6. Бахтизин А.Р. Агент-ориентированные модели экономики. М.: Экономика, 2008.
7. Макаров В.Л., Бахтизин А.Р. Новый инструментарий в общественных науках – агент-ориентированные модели: общее описание и конкретные примеры // Экономика и управление. 2009. №12 (50). – С.13-25.
8. Santa Fe Institute. URL: <http://www.santafe.edu> (дата обращения: 31.03.2011).
9. Борщев А. От системной динамики и традиционного ИМ – к практическим агентным моделям: причины, технология, инструменты. URL: <http://www.gpss.ru/paper/borshevarc.pdf> (дата обращения: 31.03.2011).
10. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с *AnyLogic* 5. СПб.: БХВ-Петербург, 2005.
11. Артеменко В.Б. Розробка електронного деканату для системи управління дистанційним навчанням MOODLE (на прикладі Львівської комерційної академії) // Тези доп. Міжн. наук-практ. конф. по вільному програмному забезпеченню FOSS LVIV-2011 (1-6 лютого 2011 р.). Львівський національний університет ім. І.Франка. Львів, 2011. – С.11-13.