

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
(ГАОУ ВО МГПУ)

Институт цифрового образования
Департамент информатики, управления и технологий

Е. Д. Патаракин

ИЗУЧЕНИЕ И МОДЕЛИРОВАНИЕ
СОВМЕСТНОЙ ДЕЯТЕЛЬНОСТИ
В СЕТЕВЫХ СООБЩЕСТВАХ

Монография

Москва
2024

УДК 37.01
ББК 71.3(2РОС)я73
П 20

*Рекомендовано ученым советом
Института цифрового образования МГПУ*

Автор:

доктор педагогических наук, доцент ***Е. Д. Патаракин***

Рецензенты:

доктор педагогических наук, профессор ***О. Н. Шилова***
(СПб АППО);

кандидат философских наук, доцент ***Б. Б. Ярмахов***
(МГПУ)

Патаракин Е. Д.

П 20 Изучение и моделирование совместной деятельности в сетевых сообществах: монография. – М.: МГПУ, 2024. – 140 с.

Монография посвящена использованию практики изучения и моделирования совместной деятельности в учебном процессе. Основу монографии составляют кейсы, полученные в ходе обучения или проведения исследований.

Монография предназначена для студентов педагогических специальностей, стремящихся глубже понять механизмы, определяющие поведение участников образовательных сетевых сообществ, таких как Wiki, Scratch и МЭШ. Изучение поведения основано на анализе следов, оставляемых участниками на цифровых объектах. Эти следы позволяют раскрыть динамику совместной деятельности.

Книга предлагает открытые инструменты для моделирования и анализа совместной деятельности, способствуя развитию эффективных методов педагогической практики в цифровом пространстве.

ISBN 978-5-243-00740-5

© Е. Д. Патаракин, 2024

© ГАОУ ВО МГПУ, 2024

ОГЛАВЛЕНИЕ

Предисловие	4
Глава 1. Науки о сетях и сообществах	6
Глава 2. Среды моделирования и изучения совместной деятельности	33
Глава 3. Примеры анализа совместной деятельности	79
3.1. Результаты библиометрического картирования поля исследования	79
3.2. Анализ совместной деятельности на вики-площадке Letopisi.org	93
3.3. Анализ совместной деятельности учителей нескольких школ внутри их школьных доменов Google Apps	95
3.4. Анализ совместной деятельности учеников в одной из студий на сайте Scrtatch	97
3.5. Совместная деятельность участников МЭШ по созданию и повторному использованию сценариев уроков	101
Глава 4. Моделирование совместной деятельности	105
Заключение	125
Литература	126

ПРЕДИСЛОВИЕ

Книга основана на реальных событиях, которые происходили с участниками совместной сетевой деятельности в сетевых сообществах.

Нами были изучены цифровые следы, которые оставляли участники совместной деятельности:

- в электронных библиотеках при создании совместных статей и разметке этих статей общими ключевыми словами;
- в различных категориях вики-проектов при создании и редактировании совместных статей;
- в студиях и коллекциях различных сообществ обучения программированию при создании и видоизменении кода проектов;
- внутри школьных доменов при совместном создании, редактировании и использовании документов, презентаций, таблиц и диаграмм;
- в цифровых образовательных репозиториях при создании, запуске, копировании и видоизменении сценариев уроков.

Основным пространством для сбора учебных и исследовательских материалов служила вики-среда, доступная на сайте: digida.mgpi.ru.

Здесь были собраны понятия, языки программирования, данные и модели, о которых рассказывается в книге. И здесь же были реализованы многочисленные учебные курсы, связанные с построением и изучением сетевых сообществ и сред многоагентного моделирования.

В монографии были использованы тексты автора, написанные в том числе в соавторстве, ранее опубликованные в научной периодике¹, а также в докторской диссертации, защищенной в Институте

¹ Патаракин Е. Д. Макроскопический подход к анализу совместной сетевой деятельности // Образовательные технологии и общество. 2017. Т. 20. № 3.

педагогического образования и образования взрослых Российской академии образования в 2015 г.¹

С. 309–329; *Его же*. Агентное моделирование для рефлексии образовательной организации // Искусственные общества. 2018. Т. 13. № 4. URL: <https://artsoc.jes.su/s207751800000133-5-1/>; *Патаракин Е. Д., Реморенко И. М., Буров В. В., Парфёнов Р. В.* Выявление ключевых участников социально-педагогических проектов // Образовательные технологии и общество. 2015. Т. 18. № 2. С. 675–692 и др.

¹ *Патаракин Е. Д.* Педагогический дизайн совместной сетевой деятельности субъектов образования: дис. ... д-ра пед. наук. СПб., 2015. 319 с.

ГЛАВА 1

НАУКИ О СЕТЯХ И СООБЩЕСТВАХ

Наука о сетях (наука о связанности) — научная дисциплина, которая изучает общие черты природных или искусственных сетей, таких как информационные, биологические и социальные сети¹. Общие подходы к изучению сетей и использованию сетевого анализа представлены в ставших уже классическими работах Д. Уоттса (Watts 2004; Watts 2003) и Л. Барабаши (Barabasi 2003). Среди работ российских исследователей необходимо отметить работы И. А. Евина (Евин 2012), А. В. Назарчука (Назарчук 2011), А. В. Олескина (Олескин 2012). Как отмечает Л. Барабаши (Barabasi 2019), сети могут существенно различаться по тем процессам, которые их порождают и формируют. «Сети клеточного метаболизма сформировались в результате эволюции, продолжавшейся миллиарды лет. Всемирная паутина строится благодаря коллективным действиям организаций и миллионов людей. Социальные сети формируются под воздействием социальных норм, чьи корни уходят вглубь тысячелетий. Несмотря на такое различие в размерах, масштабах, истории и эволюции, сети, лежащие в основании этих сложных систем, очень похожи. Архитектура природных, научных и технологических сетей подчиняется общим организационным принципам и для изучения этих систем можно использовать общий набор математических инструментов. Несмотря на то, что исследование различных сетей в науке, технологии и обществе имеет длительную историю, наука о сетях, как отдельное направление, сформировалась только в XXI веке. В XX веке не существовало инструментов для сопоставления сетей и отслеживания огромного количества данных, стоящих за этими сетями. Связанные с интернет-революцией эффективные и быстрые

¹ Наука о сетях // Поле цифровой дидактики: сайт. URL: http://digida.mgpu.ru/index.php/Наука_о_сетях (дата обращения: 02.07.2024).

методы цифрового хранения и совместного использования данных, коренным образом изменили нашу способность собирать, объединять и анализировать данные, относящиеся к реальным сетям»¹.

Термин «социальная сеть» впервые использовал британский социолог Дж. Барнс в своей статье (Barnes 1954). Существенным вкладом Барнса стало то, что он предложил не просто рассматривать взаимосвязи между людьми как сеть, а перешел к использованию конкретного инструмента — социограмм — такой формы визуального представления (диаграмм), в которой отдельные участники рассматриваемого сообщества представлены в виде узлов, отображаемых точками, а связи между ними — соединяющими соответствующие точки линиями. Определенные таким образом структуры практически полностью совпадают с понятием графа в математике, представляющего собой математическую абстракцию реальной системы любой природы, объекты которой обладают парными связями:

Графом $G = (V, E)$ называется совокупность множества вершин, что соответствует участвующим в изучаемой социальной группе лицам, и множества неупорядоченных или упорядоченных пар элементов множества, по сути являющихся множеством связей между отдельными парами индивидов.

Из данного понимания родилась новая дисциплина — социометрия — методика изучения внутригрупповых связей и иерархии в малых группах, основателем которой считается Джейкоб Морено (Moreno 1934; Морено 1958). Логика строения социальных групп, отношения членов групп между собой, а также взаимодействие между группами, описание на основе этого их структурных свойств и возникающих на их базе эффектов в дальнейшем стало основным направлением концептуализации сетевого анализа. Очевидным образом это привело к тому, что базой для анализа стало применение математических методов из области линейной и дискретной математики — теории матриц и графов.

¹ Патаракин Е. Д. Педагогический дизайн совместной сетевой деятельности субъектов образования. С. 113–114.

Сетевой анализ, используемый в социальных науках, является модификацией исследовательского инструментария, который применяется в самых разных областях: в биологии, экономике, логистике, физике и т. д. Эта модификация носит официальное название Social Network Analysis — SNA (анализ социальных сетей), однако во многих работах социальных аналитиков его обозначают просто как сетевой анализ. Это направление современной вычислительной (или, как часто, используя кальку с английского языка, говорят «компьютерной») социологии, посвященное описанию и анализу связей, возникающих в ходе социального взаимодействия (в первую очередь — коммуникации) и приводящих к образованию единой структуры — социальной сети.

Принципиально важным для сетевой науки является то, что в рамках данной дисциплины не рассматриваются индивидуальные свойства личностей (участников) или объектов, в отношении которых они действуют. При использовании методов SNA ключевым является описание характеристик, выражающих плотность, интенсивность и пространственную координацию социальных связей. Благодаря этому становится возможным выделять в рамках исследования структурные единицы в системе социальных отношений («узлы», «блоки», «клики», «кусты»). При этом в качестве результатов проведенного анализа могут быть предъявлены не только открыто обозначенные участниками рассматриваемой социальной группы взаимодействия и отношения, но и обнаруженные в результате сетевого анализа скрытые связи. Это позволяет брать в качестве объекта анализа как сеть прямых отношений участников, так и сеть опосредованных социальных взаимодействий, в рамках которой акторы, также называемые агентами, поддерживают контакт с некоторыми общими объектами и сеть связей возникает благодаря учету этих взаимодействий. В качестве акторов могут рассматриваться отдельные личности и их совокупности, например это могут быть организации. Источником такого отношения к акторам на поле социальных наук является акторно-сетевая теория, разработанная Б. Латуром и его последователями (Кнорр-Цетина 2002; Латур 2015; Латур 2014; Латур 2013;

Ло 2006). Ключевое положение теории состоит в том, что участники сетей — люди — рассматриваются наравне со всеми другими сущностями, включенными в сеть.

Сетевые модели используются для получения знания о сетевых феноменах. При анализе социальных сетей внимание уделяется связям, а не самим действующим лицам. Как правило, социальная сеть описывается графом или матрицей взаимоотношений. Предметом сетевого анализа является структура связей и отношений между людьми и объектами, входящими в разнообразные и разномасштабные общности. Традиционно для обозначения отдельного элемента социальной сети используют понятие «узел», если речь идет об исследованиях прикладного математического характера, или «актор», если подразумевается социологические исследования.

Объектом изучения акторно-сетевой теории является сеть социальных взаимодействий, неотделимая от социальных акторов. Акторно-сетевая теория обосновывает равенство всех узлов сети тем, что без других сущностей человек не может существовать ни одного мгновения. Сеть — это совокупность акторов (людей и вещей), которые взаимодействуют в едином пространстве и деятельность которых направлена на решение общей задачи. Акторы (агенты) такой сети не только связаны друг с другом в социальном пространстве, но и осуществляют совместную деятельность ради определенной цели. Сетевое равенство акторов различной природы, свойственное акторно-сетевой теории, соответствует отношениям партнерства между людьми и компьютерными программами (программными агентами). В английском языке слово «агент» обозначает помощника, выполняющего поставленную перед ним задачу. В физике и химии принято использовать термин «агент» по отношению к веществу, участвующему в химической реакции. В компьютерном мире первые упоминания термина «агент» обнаруживаются в начале 60-х годов двадцатого века в работах Д. Энгельбарта, который рассматривал отношения людей и программ как гетерогенное сообщество. Для обозначения компьютерных программ, которые могут расширить возможности группы людей

и сделать группу умнее, Д. Энгельбарт использовал термин «умные агенты» (Engelbart 1988). Компьютерные программы, которые выполняют поставленные задачи в определенной среде, называются программными агентами (Gilbert 2008). Большое внимание взаимодействию и взаимопониманию людей и программных агентов уделяет Б. Лёф (Leuf 2006). Термин «агент» в отношении компьютерной программы не просто удобное название, означающее выполнение этой программой определенной задачи. Согласно работам Б. Ривза и С. Насса, людям свойственно относиться к программным агентам как к живым, мыслящим и чувствующим существам-исполнителям (Reeves, Nass 2003).

Наука о сетях огромное внимание уделяет визуализации сетевых отношений и построению карт, которые позволяют представить и понять сетевые структуры, лежащие в основании сложных адаптивных систем. Центральная метафора теории «актор – сеть», — это метафора цикла путешествий, каждое из которых вовлекает в сеть все новых и новых акторов, увеличивая способность доминировать на расстоянии. Эту картографическую метафору используют и Б. Латур, и Дж. Ло при описании покорения земли, небес, мира микробов, экономики или бухгалтерского учета (Law 1984; Ло 2006). Концепция современного инструмента «макроскопа» для описания социологических данных была предложена Жюлем Росне (Rosnay 1979) и получила развитие и практическое воплощение в работах К. Бёрнер (Börner 2012; Börner 2011). Следует отметить, что Бёрнер и ее коллеги очень много сделали для открытости данных и дизайна систем, которые эти данные позволяют анализировать и визуализировать (Börner et al. 2020; Börner, Record 2017).

Построение карт сетевых отношений между акторами различной природы основывается на сетевых характеристиках узлов и ребер. С точки зрения формального математического описания социальная сеть, образованная акторами представляет собой граф $G = (V, E)$ в котором множество представляет конечное множество вершин (агентов), а множество $V = \{1, 2, ..., n\}$ содержит ребра, отображающие их взаимодействие. Общие параметры социальной сети представлены ниже.

Размер сети определяется максимальным расстоянием между любыми парами вершин графа. Вычисляется по формуле:

$$\theta = \max_i \max_j d(v_i, v_j),$$

где v_i, v_j любые произвольно взятые вершины графа, а $d(v_i, v_j)$ — расстояние между ними. Таким образом, размер сети может находиться в диапазоне от 1 в случае, если вся сеть состоит только из двух акторов, т. е. представлена графом из двух вершин и одного ребра, до $(n - 1)$, где n это число вершин графа.

Сетевая плотность — общая характеристика сетевой структуры, равная соотношению имеющихся и максимально возможных связей в социальной группе. Так как максимально возможное число связей в неориентированном графе равно $\frac{n \times (n - 1)}{2}$, то соответствующая плотность связей будет вычисляться по формуле:

$$\Delta = \frac{L}{n \times (n - 1) / 2} = \frac{2L}{n \times (n - 1)},$$

где равняется общему количеству связей, которые наблюдаются в рассматриваемом графе.

Показатель кластеризации характеризует степень взаимодействия между собой ближайших соседей рассматриваемого актора и представляет собой локальную характеристику сети. В большинстве сетей, если актор А соединен с актором В, а актор В с актором С, то существует большая вероятность, что акторы А и С соединены: друзья наших друзей обычно также являются и нашими друзьями. Коэффициент кластеризации данного узла есть вероятность того, что два ближайших соседа этого узла сами есть ближайшие соседи, и может быть вычислен по формуле:

$$C(i) = \frac{t_i}{q_i(q_i - 1) / 2},$$

где i — номер узла, q_i — его степень (т. е. число ближайших соседей), а t_i — число непосредственных связей между ближайшими соседями. Значение параметра лежит в диапазоне $0 \leq C \leq 1$. Если

все ближайшие соседи актора v_i попарно связаны, то $C(i) = 1$, а если между ними, наоборот, нет никаких связей, то $C(i) = 0$.

Параметр центральности актора является одним из наиболее часто востребованных при использовании методов SNA. Содержательный смысл расчета показателя централизации состоит в определении того, насколько неравномерно распределена власть / влияние в смысле, определяемом выбранными видами связей в социальной группе, представленной графом. Таким образом $C_D = 0$, тогда и только тогда, когда отсутствует преимущество в центральности любого актора, а $C_D = 1$ единственно возможно, когда доминирует единственная вершина. Показатель централизации также называют центральностью группы.

Существует несколько подходов к определению центральности. Первый, наиболее простой, говорит, что центральность актора является ничем иным, как числом связей, рассмотренным выше. Такой подход принято называть центральностью по степени. Для неориентированного невзвешенного графа это приводит к следующей формуле:

$$C_D(i) = \sum_{j=1}^n \alpha_{i,j},$$

где $C_D(i)$ — уровень центральности для актора в сети из акторов. Его максимальное значение ограничено $(n - 1)$. Вычисленная таким образом центральность позволяет сравнивать между собой акторов в рамках конкретной группы, представленной одним графом.

Следующий способ определения центральности получил название центральности по посредничеству и обозначает степень значимости акторов для потоков взаимодействия внутри изучаемой социальной сети. Центральность по посредничеству можно интерпретировать как меру контроля, осуществляемого конкретным актором по отношению ко всей сети: если у какого-либо узла высокий показатель центральности по посредничеству, можно предположить, что он — единственный связующий элемент между различными частями сети. Если кратчайший путь между вершинами v_1 и v_2

проходит по маршруту $v_1 v_3 v_4 v_2$, то акторы, обозначенные вершинами v_3 и v_4 занимают контролирующую позицию по отношению к v_1 и v_2 . Соответственно, актор тем более централен в этой парадигме, чем большее число маршрутов проходит через него (т. е. контролируется им). Это можно выразить следующей формулой:

$$C_B(k) = \sum_i^n \sum_j^n \frac{g_{ij}(v_k)}{g_{ij}}, (i \neq j \neq k),$$

где $g_{ij}(v_k)$ равно числу кратчайших путей, соединяющих пару акторов v_i и v_j и проходящих при этом через v_k , а g_{ij} — число всех кратчайших путей, соединяющих акторы v_i и v_j .

В последние годы активно развивается наука о командных научных исследованиях (Science of Team Science). Эта область представляет одно из направлений гуманитарных вычислительных наук, цифровой и сетевой социологии (Берроуз, Севидж 2016; Журавлева 2015; Мещерякова 2020). «Командная наука» исследует поведение людей в современных командах в сложных технологических средах, которые во многом определяют то, как протекает совместная работа и совместное обучение. При этом для анализа положения конкретного агента, как правило, используются сетевые характеристики, перечисленные выше. Степень центральности показывает, кто является наиболее активным узлом в сети. Измеряется количеством связей с другими узлами в сети. Центральность (в любой из интерпретаций) показывает, насколько данный узел близок по отношению к другим узлам в сети. В соответствии с теорией сетей большое количество взаимодействий узла может не только изменить позицию узла в сети, но также и изменить позиции других узлов. Центральность по посредничеству выражает, сколько кратчайших путей между всеми узлами сети проходит через определенный узел, т. е. это мера степени контроля сети узлом. Она характеризует, насколько важную роль данный узел играет на пути между другими узлами. Если у какого-либо автора в сети научных публикаций высокий показатель центральности по посредничеству, можно предположить, что он связывает авторов, принадлежащих к различным научным командам.

При анализе командных игровых действий центральность по посредничеству позволяет судить, насколько взаимодействие между двумя игроками зависит от третьего игрока. Игроки с высоким уровнем центральности по посредничеству играют ключевые роли в поддержании темпа игры. Если речь идет о контроле над информационными потоками в социальной сети и степени влияния на других, то для такого контроля актор должен быть посредником между другими узлами, поскольку это дает ему возможность прервать контакт между ними. Именно центральность по посредничеству является наиболее подходящей мерой для определения степени способности индивида контролировать взаимодействие людей в своем социальном окружении. Центральность по посредничеству является глобальной характеристикой вершины. Показатель учитывает лишь кратчайшие пути от вершины к вершине и основан на предположении, что при наличии между двумя вершинами нескольких коротких путей равной длины каждый из них используется с равной вероятностью. Индекс информационной центральности оценивает распределение информации во всех цепях графа, взвешивая цепи величиной, обратной их длине. Чем больше возможности участника сетевого взаимодействия для распространения информации, тем выше его эффективность в ее распространении.

Исходя из этого, мы можем исчислить интегрированный показатель для каждого узла или модуля сетевого сообщества. Следует отметить, что показатель сетевой центральности по посредничеству использовался в целом ряде работ при определении ключевых игроков в различных контекстах совместной деятельности (в социальных сетях, террористических группах, командных играх). Независимо от контекста исследования ключевыми игроками являются участники с наибольшими показателями сетевой центральности по посредничеству, способные контролировать информационные потоки, эффективно распространять информацию и обладающие наибольшим социальным капиталом. Как правило, прикладные задачи выявления ключевых игроков связаны с их дальнейшим использованием или нейтрализацией.

Научный подход предполагает, что несколько человек работают на достижение общей научной цели. При этом размер команд может варьировать от пар участников (диад) до крупных команд, в состав которых входят более мелкие команды. Научные команды собираются для выполнения конкретных проектов и в них могут входить сотрудники из разных организаций, расположенных в разных местах. Эти команды отличаются от формальных академических организаций, таких как отделы или лаборатории, которые иногда называют командами.

Следует отметить, что развитие командной науки (Team Science) стало возможным благодаря накоплению огромных массивов библиографических данных и разработке инструментов, которые сделали работу с такими данными доступными для широкого круга исследователей. В отношении неявных научных групп разработаны методы исследования, как правило, включающие библиографический и сетевой анализ, которые приводят к выделению графов, объединяющих ученых, принадлежащих к одной группе. Примеры выделения команд ученых, работающих над общей тематикой, приводятся в третьей главе монографии. Сходные методы используются в организационном сетевом анализе, когда внутри организации выявляются группы людей, которые интересуются общими темами, обмениваются документами, просматривают документы друг друга и даже редактируют общие документы. И тот же подход можно использовать при изучении обучающихся в образовательных сообществах, где они могут формально относиться к разным группам и разным школам, но сотрудничать в написании общих статей, видоизменении проектов, компьютерных моделей или сценариев уроков. Возможность наглядного представления результатов извлечения знаний из исходных данных (Whitney 2012), которую обеспечивает компьютерная аналитика, привлекает внимание специалистов различных областей знаний. Особое значение имеет аналитика с позиций науки о сетях, которая рассматривает мир физических и социальных систем как сеть гетерогенных акторов, связанных между собой различными типами связей (Luke 2015). Наука о сетях и методы сетевого анализа активно

используются для исследования различных областей совместной деятельности. Поиску взаимосвязанных групп — клик и отношения между центральными и периферическими узлами сети, определению позиций, которые занимает тот или иной участник в сетевой структуре, посвящены работы У. Брандеса (Brandes 2020; Brandes et al. 2016; Brandes, Erlebach 2005). В последние годы были опубликованы многочисленные работы, связанные с использованием сетевого анализа в археологии (Donnellan 2020; Peeples 2019; Sporns 2011). Сетевая наука, сетевое образование и сетевой анализ все чаще используются в вопросах здоровья и медицины (Margolis, Parboosingh 2015). В обзорной работе Х. Веше и соавторов (Wäsche et al. 2017) авторы рассматривают возможности сетевого анализа применительно к современному спорту. Особенно показательно успешное приложение методов сетевого анализа к области спортивных командных игр (Dechurch, Mesmer-Magnus 2010; Gama et al. 2014; Kröckel et al. 2017). На наш взгляд, эти достижения могут быть перенесены в область совместных образовательных практик, где достижения методов сетевого анализа пока скромнее (Huerta-Quintanilla et al. 2013).

Близкий к «командной науке» подход развивается в рамках направления сообщества практики, которое представляет собой множество субъектов, объединенных общими интересами, объектами и общей продуктивной деятельностью. Внутри сообществ практики участники постепенно переходят от позиции наблюдателя к позиции активного пользователя, а от позиции пользователя к позиции дизайнера и создателя собственных учебных сценариев. Это направление опирается на работы Э. Венгера (Wenger 1999; Wenger et al. 2002), в которых хотелось бы выделить понятие граничных объектов (*boundary objects*), способных одновременно принадлежать к разным практикам и использоваться разными людьми и разными сообществами по-разному (ср.: понятие «социальный объект» (Engeström 2005)).

К. Кнорр-Цетина предложила для описания особой формы отношений, которая складывается между людьми и объектами, теорию объектно-центрированной социальности (*object-centered sociality*)

(Кнорр-Цетина 2006). Отталкиваясь от нее, финский исследователь и инженер Ю. Энгестрём ввел в использование понятие «социальный объект» (Engeström 2005). Ю. Энгестрём подчеркивает, что основой для формирования отношений и связей между людьми в современных сетях на основе социальных сервисов служат не общие формы деятельности, а конкретные объекты, выступающие в роли социальных катализаторов. Для объяснения понятия «социальный объект» исследователь использует метафору футбольного мяча, появление которого на пляже приводит к тому, что вокруг него собирается группа детей. Ранее эту же метафору использовал Б. Латур, описывая утверждение, которое постоянно находится в опасности остаться потерянным, как мяч при игре в регби (Латур 2013). А еще раньше метафору мяча использовал М. Серр, описывая формирующий коллектив социальный квазиобъект (Serres 2007). Во второй главе мы несколько раз используем метафору мяча на футбольном поле применительно к социальному объекту, прикасаясь к которому участники оставляют на нем свои следы.

При этом современные социальные объекты являются цифровым объектом. Цифровой объект состоит из структурированной последовательности байтов, имеет название, уникальный идентификатор и атрибуты, описывающие его свойства. Основные характеристики цифрового объекта — редактируемость (изменяемость), интерактивность, открытость, копируемость и распространяемость. Цифровой объект можно повторно использовать в новом учебном контексте, и его не нужно привязывать к дисциплине или предмету, для которого он был создан. Это позволяет креативно использовать цифровые объекты, включая их в различные контексты образования (Kallinikos et al. 2013; Kallinikos and Mariátegui 2011; Wang 2008). Простые в использовании и легко создаваемые цифровые объекты открывают новые горизонты для образования. И это связано не только с возможностью разнообразных совместных действий по созданию и повторному использованию цифровых объектов, но и с возможностью изучения деятельности на основании цифровых следов, которая эта деятельность оставляет на цифровых социальных объектах.

Сравнение известных сетевых учебных сообществ позволяет увидеть, что практически все они используют и в явном виде проговаривают для участников идею спирали действий, которые совершают люди — агенты деятельности — над цифровыми объектами деятельности. Наиболее отчетливо эта спиральная последовательность действий была сформулирована М. Резником для сообщества Scratch: «Вообрази -> Создай -> Играй -> Делись -> Обсуди -> Вообрази» (Resnick 2017). Естественно, что сходные последовательности действий над объектами были предложены для близких сред обучения программированию, где социальным объектом является код проекта, который всегда можно посмотреть, выполнить и передать сообществу для дальнейшего использования. Однако мы видим, что практика повторных действий не ограничивается только программным кодом проектов. Сходным образом организована деятельность участников в сообществе GeoGebra, где они делятся апплетами учебных приложений (Hohenwarter, Lavicza 2011); в сообществе WebGrid обмениваются репертуарными решетками и результатами исследований; в сообществах Letopisi и Preobra — фрагментами текстов; в сообществах CloudWorks и МЭШ — сценариями уроков (табл.).

При разработке сетевых моделей совместной сетевой деятельности мы исходили из того, что всякое действие субъекта над объектом приводит к образованию связи между субъектом и объектом. Если субъекты деятельности совершают действия над одним и тем же объектом, то этот объект выполняет связующую функцию и является социальным объектом. Различные формы совместной сетевой деятельности (совместное редактирование документа, создание и комментирование записей в блоге, создание и совместное редактирование статей в вики и т. п.) могут быть сведены к единой схеме, которая позволяет анализировать и сравнивать деятельность участников.

В онлайн-среде основным видом сотрудничества являются ремиксы. Пользователи создают новый проект на основании существующих проектов, которые другие пользователи ранее опубликовали в сообществе. Например, более 27 % всех Scratch-проектов представляют собой ремиксы других проектов и пользователи

Многообразие действий над объектами в учебных сообществах

Сообщество	Объекты	Действия	Доступность объектов
Scratch scratch.mit.edu	Scratch-проект	Вообрази -> Создай -> Играй -> Делись -> Обсуди -> Вообрази	Каждый опубликованный проект может быть использован другими участниками для создания нового проекта. Создание ремиксов поощряется
NetLogo Modeling Commons modelingcommons.org	Модель NetLogo	Создай -> Экспериментируй-> Делись -> Обсуждай -> Изменяй -> Создай новый вариант	Автор модели может пригласить других участников. Эти сотрудники получают право создавать свои версии модели. Участники могут обсуждать модели и добавлять к ним теги
StarLogo www.slnova.org	Трёхмерная модель StarLogo TNG	Создай -> Экспериментируй -> Редактируй -> Делись	Каждая модель может быть скачена, изучена и видоизменена другими участниками
Looking Glass lookingglass.wustl.edu	Alice — анимированная трёхмерная история	Создай -> Анимируй -> Делай ремикс -> Делись	Созданные одним автором шаблоны и анимации могут быть извлечены и использованы другими авторами внутри их собственных моделей
GeoGebra https://www.geogebra.org	Апплет GeoGebra	Найди -> Используй -> Видоизмени -> Поделись	Динамические чертежи могут повторно использоваться и видоизменяться как ремиксы
CloudWorks cloudworks.ac.uk	Карта учебного дизайна (Compendium LD)	Найди -> Поделись -> Обсуди	Карты учебного дизайна могут обсуждаться, оцениваться и видоизменяться другими участниками

Сообщество	Объекты	Действия	Доступность объектов
WebGrid webgrid. typed.com	Репертуарная решетка	Покажи -> Кластеризуй -> Построй карту -> Сравни	Репертуарная решетка, созданная одним автором, может заполняться другими участниками. Решетки можно сравнивать или объединять
Letopisi.org	Вики-страница	Создай -> Редактируй -> Обсуждай	Каждая созданная страница может обсуждаться и дополняться любым из участников
Преобра.ru	Фрагмент документа	Оцени -> Обсуди -> Предложи свой вариант	По каждому опубликованному фрагменту участники могут предлагать свои версии
https://uchebnik.mos.ru	Сценарий урока	Оцени -> Загрузи -> Копируй -> Запусти -> Встрой в задания	Для каждого опубликованного учебного сценария учителя могут создавать свои ремиксы. Создание ремиксов поощряется

уже выложили более 27,7 млн проектов в Scratch-сообществе (Monroy-Hernández 2013). Сообщество, использующее ремиксы для обучения программированию, опирается на принципы теории конструкционизма, согласно которой мы учимся быстрее, проектируя и создавая объекты, и это обучение становится еще более значимым и эффективным в социальном контексте (Dasgupta et al. 2016).

Большое значение для организации совместной деятельности имеет поддержка формирования групп в цифровой среде. Здесь интерес представляет обзорная работа, в которой авторы проанализировали совместную деятельность группы, команды, объединения и формирования при поддержке компьютеров Computer Supported Cooperative Work (CSCW) в период с 1992 по 2018 гг., сформулировав запрос к системе «цифровая среда» (Gómez-Zarà et al. 2020). Авторы изначально собрали более 2000 статей, из которых отобрали 35, сосредоточившись на технической поддержке формирования групп. Авторы подчеркивают значение субъектности и самодетерминации деятельности участников, которые сами принимают решение о том, в какие группы им вступать. В работе прослеживается несколько периодов, когда исследователей интересовали различные особенности групп и размерности групп, от совсем маленьких диад к командам, группам, толпам и сообществам.

В представленной авторами типологии различаются следующие типы объединений участников:

Диады — парные объединения, которые возникают по взаимному согласию участников и могут возникать при помощи экспертных систем, советующих обратить внимание на другого участника. Возможный пример механизма образования диады — приглашение к работе над общей моделью в среде NetLogo, запросы на помощь в решении задачи на StackOverflow или WikiHow. Минимальное объединение, которое легко образуется, но также легко и распадается.

Команды — объединения нескольких участников, работающих над общей задачей. Объединение участников происходит самопроизвольно и в современной цифровой среде поддерживается огромным разнообразием программного обеспечения, которое

помогает им совместно редактировать тексты, тексты программ, ментальные карты, диаграммы, аудио- и видеофайлы. Общая деятельность предполагает наличие или формирование у участников общей экспертной позиции.

Группы — объединения участников, которые могут носить закрытый характер и поддерживаться множеством социальных сетей. Как правило, включение в группу предполагает общие интересы и, возможно, общую позицию. Близкие по принципу устройства социальные сети, в которых участники отслеживают события, новости, ссылки, которые публикуют во френдленте те, кого они выбрали в друзья: «Живой Журнал», Google+, «ВКонтакте», «Одноклассники». Участник настраивает канал (ленту) своих сообщений. Лента может быть общедоступной или приватной. На записи, которые публикует участник, можно подписаться. Все записи участника (ссылки, фотографии, видео) публикуются в его ленте. Записи тех, на кого участник оформил подписку, публикуются во френдленте этого участника. К открытым записям могут добавлять лайки и комментарии любые участники сети. Существует возможность создания открытых и закрытых тематических групп, в которых могут публиковаться записи по заданной тематике. Правила регистрации в группе и правила поведения в группе определяет тот, кто ее создал, или те, кому он делегировал права администрирования. Организационная структура выстраивается вокруг общения участников. В системе прямые связи (отношения) между участниками и эти отношения определяют, что видят участники в своих каналах (лентах) сообщений. Отдельный интерес для нашего исследования могут представлять социальные сети, специально поддерживающие образование исследовательских групп и формирование структур «невидимого колледжа». К таким исследовательским социальным сетям относятся: Academia, Mendeley, ResearchGate. Участник получает возможность публиковать в системе свои исследовательские работы. Кроме того, программа отслеживает публикации данного автора в электронных изданиях и предлагает добавить их в базу данных. Программа позволяет находить авторов, пишущих на близкую

тему. Стандартный для социальных сетей механизм подписки на действия других участников. После подписки участник видит в своей френдленте, что опубликовали или отметили закладкой те, на кого он подписался.

Толпы — множества участников, как правило, конкурирующих за решение предложенной к выполнению задачи. Как правило, организаторы краудсорсинговых проектов стремятся собирать индивидуальные решения и предложения, но часто наблюдаются случаи самоорганизации толп в группы. Примером может служить ZooUniverse — сеть решения научных проблем, где участники совместно сортируют объекты. Администратор помогает организатору исследования разместить данные на площадке. Как правило, это цифровые фотографии (древние манускрипты, массы глубоководного планктона, участки поверхности далеких планет, звездное небо). Участникам предлагается решить простую задачу — отнести наблюдаемые объекты к какому-либо классу. Решение находится благодаря тому, что в работе принимают участие множество людей, способных к разделению визуальных образов. Как правило, эксперты не только размещают данные, но и снабжают их подробными справочными материалами, на основании которых участники осуществляют процедуру разделения объектов. Бывают модификации, когда для решения самой задачи по разделению объектов регистрация в системе не требуется. В этом случае зарегистрированный участник должен принять обоснованное решение о том, насколько незарегистрированные участники справились со своей задачей. Деятельность зарегистрированных участников контролируется экспертами. Никаких явных связей, которые они сами могли бы проследить, между участниками в ходе этой деятельности не возникает. Российские примеры организации краудсорсинговых проектов, в которых мы наблюдали самопроизвольное образование групп, члены которых выражали общую экспертную позицию, были связаны с использованием площадки для менеджмента идей (Патаракин и др. 2012). Возможности площадки позволяли участникам группы, покупать игровые акции и определять будущее идеи не только голосами, но и вложенными

в развитие идеи виртуальными деньгами. Цель проекта — сформулировать и отобрать лучшие предложения по продуктивному использованию в российских школах интернет-практик. Анализ предложений, которые выдвигали учителя школ и преподаватели вузов на площадке для выработки перспективных интернет-практик, позволил выделить несколько наиболее перспективных направлений для развития повсеместного обучения. При этом были отмечены многочисленные примеры того, как участники объединялись в группы и высказывали не индивидуальную, а групповую позицию.

Сообщества — множества участников, которые свободно собираются на поле совместной деятельности и в ходе деятельности дифференцируются по степени вклада в общую деятельность, правам и статусу. Существование сообществ поддерживается разнообразными программными средствами, которые позволяют участникам выстраивать отношения внутри сообщества. Сюда можно отнести вики («Википедия»), системы QA и GitHub — сеть программистов с возможностью хранить репозиторий проекта и все его изменения.

Наибольшую известность и популярность среди систем совместного создания контента получили вики- и википодобные системы коллективного редактирования гипертекстов, где каждый автор может создавать и редактировать любую страницу. Действия над страницей разнообразны: проверить содержание, поправить форматирование, добавить иллюстрации, классифицировать страницу при помощи категорий. Вся история изменений записывается в истории версий. Участники редактирования страницы могут договариваться между собой, обмениваясь сообщениями либо в пространстве личных страниц участников, либо в пространстве обсуждений страниц. Все связи участников опосредованы страницами, которые они редактируют. Организационная схема деятельности вики системы представлена на рисунке 1.

Общий анализ функционирования QA-систем представлен в работе Ванга и соавторов (Wang et al. 2014). Сравнительный анализ возможностей и особенностей StackOverFlow и Stack Exchange сделан в работе Оливейры (Oliveira et al. 2018). Проект

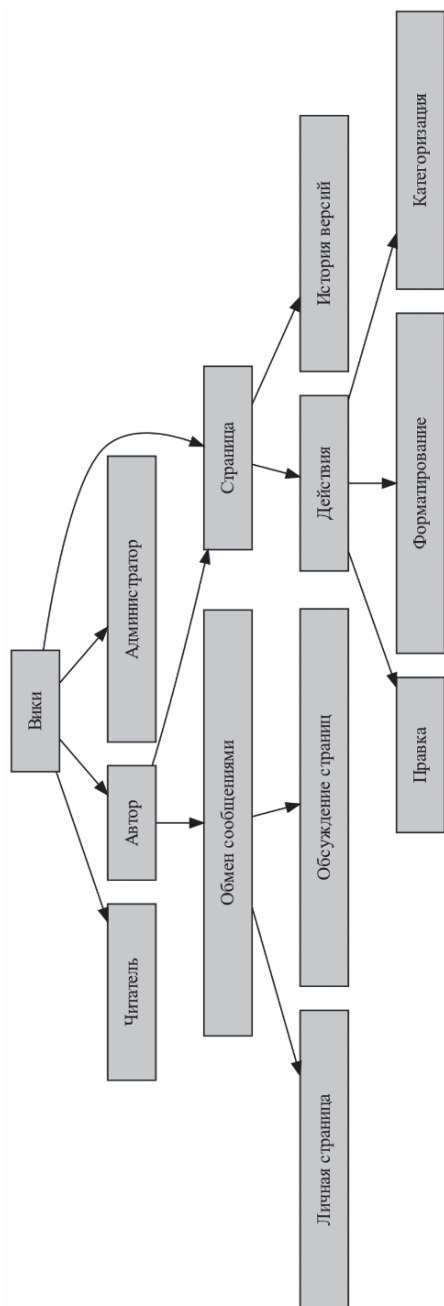


Рис. 1. Схема вики-системы

StackOverFlow как сеть создания контента в форме вопросов и ответов о программировании стартовал в 2008 году и объединяет около 7 млн программистов.

Практическим результатом деятельности сообщества является пополняющаяся коллекция знаний, сформулированных как ответы на вопросы. По сути дела, в ответах каждый раз представлены варианты сценария достижения нужного результата или решения проблемы, с которой столкнулся автор вопроса. Довольно часто вопрос формулируется в форме *«Я пытаюсь получить результат, но у меня не получается»*. При этом система настроена так, что автор вопроса сначала ищет ответы среди тех вариантов, которые уже сформулированы внутри системы. В дальнейшем при настройке собственной экспериментальной площадки мы использовали этот подход в категории «Рецепты», где собраны примеры того, как может быть решена та или иная проблема. При этом личные учебные результаты учеников могут быть связаны как с поиском ответов (умение сформулировать проблему, умение воспроизвести решение в новой среде), так и с ситуациями, когда они выступают в роли экспертов и фиксируют внутри системы свой собственный опыт. Организационная схема представлена на рисунке 2.

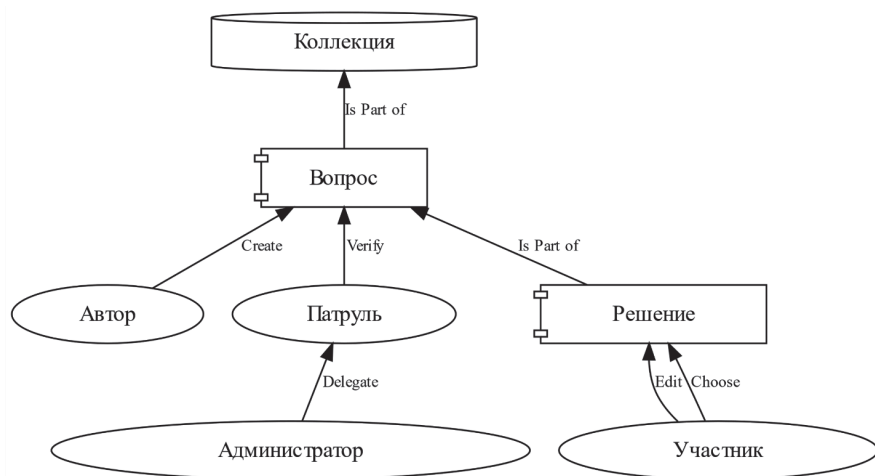


Рис. 2. Схема сообщества StackOverFlow

GitHub — сеть программистов с возможностью хранить репозиторий проекта и все его изменения (Payette 2020; Воинов и др. 2019). Участник получает возможность создавать репозиторий, в котором собираются материалы проекта. Для репозитория обязательно наличие файла ReadMe, где описываются цели и планы проекта. Программа-интерфейс GitHub поддерживает создание новых версий в репозитории при сохранении всех предыдущих версий (так же, как и вики). Автор (хозяин) репозитория может дать доступ и права к созданию новых версий в своем репозитории членам своей рабочей группы. В этом случае они создают новые версии, которые вливаются в репозиторий. Для открытых проектов, а все бесплатные проекты на GitHub открыты по умолчанию, для любого участника существует возможность создания новой собственной версии репозитория. В этом случае такой участник становится новым хозяином и может развивать проект как собственный. Взаимодействие участников выстраивается на основе продукта (документа). Обобщенная диаграмма отношений участников сообществ обмена кодом представлена на рисунке 3.

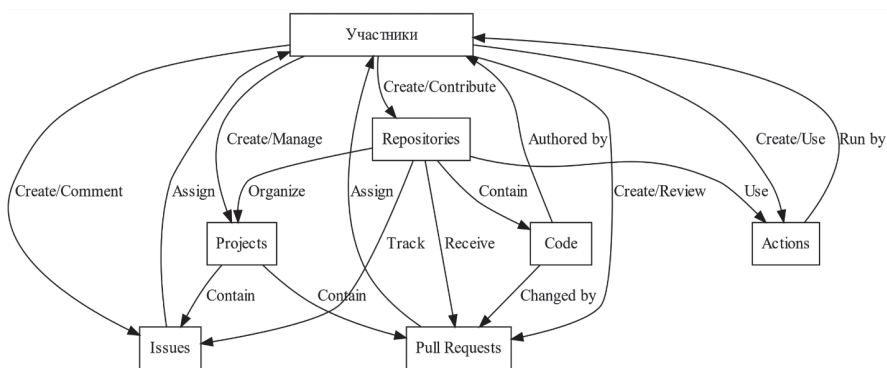


Рис. 3. Диаграмма систем обмена кодом

На основании общих, но видоизмененных принципов GitHub построены и разнообразные репозитории обучающего кода в таких системах совместной деятельности, как Scratch, Snap!, StarLogo Nova, NetLogo и целый ряд других. Основные действия участников:

размещение (публикация) кода, создание версий, оценивание, комментирование. Сходные организационные принципы используются в образовательных цифровых репозиториях. Например, в репозитории учебных и исследовательских моделей ComSes (Rollins et al., 2014; Vendome et al., 2020), в котором авторы могут публиковать свои статьи, добавляя к ним модели. В качестве основных средств программирования здесь используется NetLogo, Julia, R. Еще один пример из категории «код» — Courseware as Code — организация учебных курсов на основе выполняемых скриптов (Rodriguez et al. 2018; Rodriguez et al. 2020) и хранение материалов в GitLab (Choudhury et al. 2020; Safari et al. 2020). В категорию «код» попадает международный проект Fusepool (Kaschesky, Selmi 2013), направленный на разработку компонентов программного обеспечения для представления открытых данных и сбора лучших практик использования таких данных студентами. Практическим результатом деятельности сообщества является развитие кода — создание новых возможностей, новых моделей, если предметом программирования является создание моделей. Личные результаты учащихся зависят от освоения умений программирования и моделирования. Связанное с созданием программ и моделей направление развития совместной деятельности представляется наиболее перспективным на поле цифровой и вычислительной дидактики. Мы уделили ему особое внимание при организации экспериментальной площадки, чтобы участники могли обмениваться не только текстами, но и результатами выполнения программ.

Компьютерное моделирование и особенно многоагентное моделирование открыли перед социальными науками принципиально новые возможности исследования социальных процессов. Наиболее ярко новый подход был определен Дж. Эпштейном как генеративная или вычислительная социология, задача которой состоит в том, чтобы вырастить феномен, который исследователь пытается понять. В. Л. Макаров (Макаров 2012) в своих работах о построении искусственных сообщества пишет, что ключевой фразой новой методологии многоагентного моделирования можно считать цитату из книги *Growing Artificial Societies* (Epstein, Axtell

1996, 33): «Однажды люди применительно к социальному явлению вместо вопроса “Можете ли вы это объяснить?” спросят «Можете ли вы это построить (вырастить)?». Текст книги Эпштейна и Акстелла выстраивался вокруг компьютерной модели, в которой они анализировали поведение жуков в сахарном ландшафте.

Имеется некоторая окружающая среда — сахарный ландшафт, — где в двумерном пространстве разбросан сахар — где-то больше, где-то меньше; и туда же помещены агенты-жуки, которые ползают по сахарному ландшафту по простым правилам: агенту надо есть сахар, и он перемещается туда, где сахара больше. Свойства агента задаются двумя параметрами: горизонтом видения и уровнем метаболизма, т. е. количеством сахара, которое нужно съесть, чтобы остаться в живых. При этом агент может носить с собой сколько угодно сахара. Если сахара недостаточно, агент умирает. Правила поведения агентов:

- 1) осмотреть все доступные для агента окрестности в четырех или восьми направлениях (только по прямой или по прямой и по диагонали) и выбрать незанятую клетку с максимальным количеством сахара;

- 2) передвинуться в эту клетку и собрать весь имеющийся в ней сахар.

Агенты двигаются, поедают сахар, который появляется в той или иной точке в соответствии с определенными правилами. В дальнейшем модель стали усложнять, и следующие ее варианты превратились в настоящее искусственное общество. Например, экономика появилась в «сахарной модели» после того, как в нее добавили сироп. Жуки стали есть сахар и сироп, и у одних оказалось больше сахара, у других сиропа, они начали встречаться и обмениваться субстратами, конечно, соблюдая определенные правила обмена. Какое-то количество сахара можно обменять на некоторое количество сиропа — возникает цена.

Ключевое утверждение генеративной социологии состоит в том, что исследователь может доказать свое понимание события, только смоделировав и вырастив его в контролируемой агентной среде. Формулировка Дж. Эпштейна «понять значит суметь вырастить»

очень близка формулировке А. П. Ершова, который в работе «Программирование — вторая грамотность» писал: *«в своей творческой природе программирование идет намного дальше большинства других профессий, приближаясь к математике и писательскому делу... Один из тезисов современной теории познания гласит “мы знаем это, если можем это запрограммировать”»* (Ершов 1981).

В настоящее время агентное моделирование активно используется для исследования социальных процессов в различных сферах. Среди основных направлений использования агентного моделирования следует выделить моделирование экологии, экономики и экономической географии, моделирование процессов обучения, спортивных командных соревнований, политических процессов. В России использование агентного моделирования связано с направлением «искусственное общество» и работами В. Л. Макарова (Макаров, Бахтизин 2015; Макаров, Бахтизин 2009).

Поскольку моделирование, как правило, связано с изучением сложных и комплексных систем с огромным количеством участвующих агентов или узлов, то содержательная область агентного моделирования пересекается с наукой о сетях и сложных системах (Cramer et al. 2015; De Caux et al. 2014; Sayama 2015; Sayama et al. 2016; Youngman, Hadzikadic 2014). Вследствие этого перспективным представляется использование многоагентного моделирования вместе с сетевым анализом (Fontana, Terna, 2015) in the complex systems framework. Such models can be usefully employed at two different levels: to help in deciding (policy-maker level). Методы имитационного многоагентного моделирования все чаще используются в качестве инструмента исследования на поле сетевой науки. Научный прогресс в современном мире все больше зависит от использования имитационного моделирования, которое служит инструментом исследования и образования. Имитационное моделирование позволяет создавать искусственные системы и цифровые двойники, которые можно запускать и исследовать с различной скоростью (Lorig, Timm 2020).

Генерация данных в многоагентных компьютерных моделях является результатом исследовательской деятельности по отношению

к предметам и явлениям, структура и образ поведения которых воссоздается в модели. Разнообразие сред агентного моделирования, для систематизации которых были разработаны различные типологии, проанализированы в работе 2020 года (Rakić et al., 2020). Среди открытых исследовательских сред ведущие позиции занимает среда NetLogo, которая открывает свои возможности не только для исследователей, но и для учащихся. В этом плане показательны две недавние книги, где представлены и исследовательские, и образовательные возможности среды (Railsback, Grimm 2019; Wilensky, Rand 2015).

Развитие сетевой науки, активное и успешное использование методов сетевого анализа и агентного моделирования в различных областях знания привело к тому, что в образовании появился запрос на сетевую грамотность учеников. Сетевая грамотность — базовые знания и умения, необходимые человеку для успешной деятельности в современном мире сложных систем и больших данных (Brandes 2020; Brandes et al. 2016; Brandes, Erlebach 2005). Ученые, придерживающиеся понятия «сетевая грамотность», активно разрабатывают учебные ресурсы, способствующие формированию ключевых навыков XXI века у учеников. Эти навыки включают способность искать, выявлять и анализировать сетевые закономерности в окружающих системах, визуально отображать данные с помощью компьютерных программ, понимать изменения в моделях и их роль в поддержке исследовательских навыков (Cao et al. 2022; Cao et al. 2023; Sayama et al. 2016; Sheetz et al. 2015).

Знакомство учащихся и преподавателей с сетевыми науками и большими данными может начаться с изучения карт, основанных на данных из разных областей исследований. Эти карты приглашают учащихся увидеть и понять сетевую науку (Börner et al. 2009). Данные для построения карт могут быть либо взяты из открытых источников, либо порождены в результате действий множества искусственных объектов, населяющих искусственное сообщество. При этом мы можем сочетать методы сетевого анализа и методы агентного моделирования (Fontana, Terna 2015). Как писал Дж. Эпштейн в своей трилогии (Epstein 2013;

Epstein 2006; Epstein, Axtell 1996) о порождающей социальной науке, моделирование искусственного общества позволяет нам развивать социальные структуры и демонстрировать, что определенные наборы простых правил достаточны для формирования сложных феноменов.

Представление системы совместной сетевой деятельности как сети отношений открывает перед педагогической наукой дополнительные возможности, связанные с привнесением методологического подхода, который показал свою эффективность в анализе феноменов различных областей знаний — физики, химии, социологии. Базовые понятия — сеть, сообщество, агенты, узлы и связи — с успехом могут быть использованы при описании ситуаций учебной сетевой деятельности. Поскольку связи цифровых объектов и людей, которые создают, изменяют и используют эти цифровые объекты, можно наблюдать, анализировать, исследовать и моделировать при помощи компьютеров, мы получаем в распоряжение мощный исследовательский комплекс, который с успехом может быть использован в педагогическом дизайне совместной сетевой деятельности (Patarakin, Shilova 2015). Мы рассматриваем совместную сетевую деятельность не только как деятельность в информационной среде при помощи сетевых сервисов, но и как развитие сетевой системы, объединяющей множество взаимосвязанных участников, объектов, которые они используют, и продуктов, которые они создают. Множество участников и объектов не просто располагаются в сетевом пространстве, они являются частью сетевой системы, которая развивается благодаря индивидуальным и коллективным действиям. В процессе совместной сетевой деятельности создаются и видоизменяются цифровые объекты. Участники совместной сетевой деятельности всегда объединены общими объектами (статьями, фотография, диаграммами, программами), в отношении которых они совершают свои действия.

ГЛАВА 2

СРЕДЫ МОДЕЛИРОВАНИЯ И ИЗУЧЕНИЯ СОВМЕСТНОЙ ДЕЯТЕЛЬНОСТИ

Существует разнообразие сред агентного моделирования, для систематизации которых были разработаны различные типологии. Недавно авторы проанализировали семь обзоров средств агентного моделирования, которые были опубликованы в последние годы (Rakić et al., 2020). В наиболее полной работе описаны 85 сред агентного моделирования (Abar et al., 2017). Среди разнообразия сред агентного моделирования существует несколько, использование которых является определенным сложившимся к настоящему времени стандартом. Так, среди открытых исследовательских сред ведущие позиции занимает NetLogo, что связано и с хорошей репутацией, и историей применения языка в научных исследованиях, и с тем, что он достаточно легко осваивается непрофессиональными программистами. Для трехмерного моделирования с целью демонстрации возникающих феноменов в образовательных целях могут использоваться многочисленные агентные среды, предназначенные для обучения вычислительному мышлению и программированию, начиная от самых простых сред визуального программирования — Scratch, Snap! StarLogo Nova и других. Выбирая среду многоагентного моделирования, мы опирались на собственный опыт сравнительного описания учебных сред (Парандекар и др. 2019), среди которых многие являлись средами агентного моделирования.

В книге «Переворот в сознании» Пейперт писал о необходимости наличия *«объекта, при помощи которого можно лучше думать»*. В поведении такого объекта ученик может увидеть свое собственное поведение. Впервые этот подход был воплощен в языке Logo, когда ученик получил в свое распоряжение агента — Черепашку, которой можно было давать простые команды: вперед, назад, направо, налево, повтори и т. д. Из этих простых

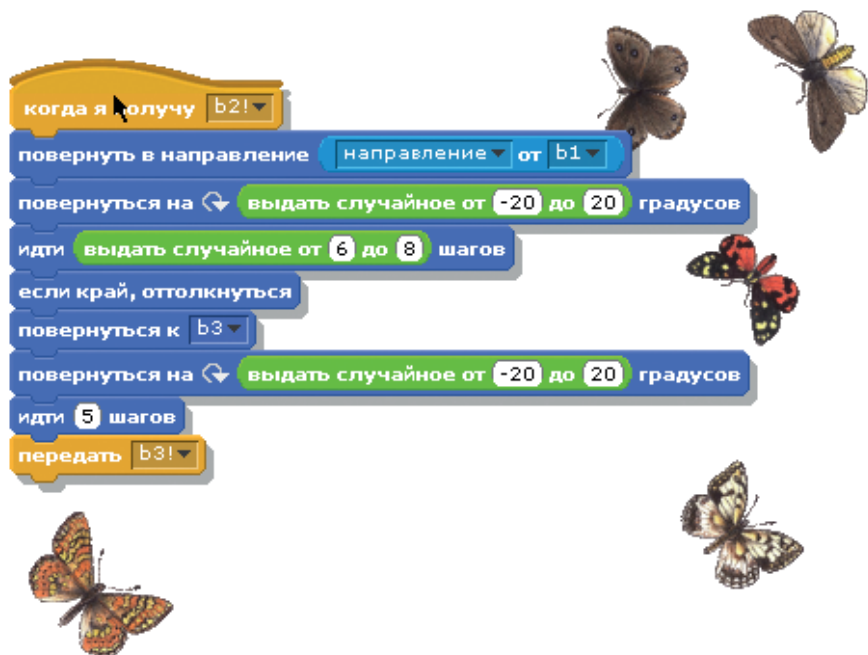
кирпичиков постепенно складываются значительно более сложные микромиры. Основной методологический принцип продуктивной субъектности, который развивал С. Пейперт и его последователи, состоит в том, что *«обучение происходит наиболее эффективно, если учащийся вовлечен в создание общественного объекта (public entity), будь то замок из песка, машина, книга или компьютерная программа»* (Harel, Papert 1991).

Визуальное программирование было создано для решения проблемы освоения учениками сложного синтаксиса программы. Визуальное программирование упрощает интерфейс, устраняет связанные с синтаксисом проблемы и позволяет новичку сосредоточиться на развитии вычислительного мышления. Абстрагирование от синтаксиса оказалось мощным стимулом. Теперь новички могут с первых дней писать сложные и эффективные программы, что является одним из требований концепции «низкий пол, высокий потолок» для преподавания программирования детям. Языки визуального программирования, такие как Scratch, Snap! и StarLogo Nova, являются практической альтернативой сложным языкам программирования с текстовым синтаксисом. Они существенно упрощают синтаксис и делают команды более похожими на разговорный язык. Они заменяют слова картинками или блоками.

Например, в Scratch блоки, похожие на кирпичики Lego, представляют несколько управляющих конструкций. «Программирование» осуществляется путем перетаскивания и составления этих блоков друг с другом, как в пазлах (Ruf et al. 2014). Как и кирпичики Lego, блоки соединяются друг с другом только в правильном порядке. Синтаксис утверждений закодирован в блоках, и учащимся нужно всего лишь распознать блоки, что не позволит им создать утверждение с синтаксическими ошибками.

Для организации учебных курсов по изучению и моделированию взаимодействия агентов внутри сетевых сообществ важно, что мы можем внутри среды блочного визуального программирования организовывать обмен сообщениями между агентами, запоминать положение и направление агентов на экране. Например,

в следующем примере летающие агенты — бабочки — обмениваются сообщениями и действуют на основе этих сообщений (рис. 4).



Источник: здесь и далее на рисунках представлены скриншоты из среды программирования.

Рис. 4. Взаимодействие агентов в среде Scratch

Мы предлагаем этот проект в качестве одного из самых первых и простых в курсе изучения сетевых сообществ, чтобы учащиеся могли сделать ремикс проекта Butterfly Effect (<https://scratch.mit.edu/projects/293659/>) и видоизменить его, увеличив число взаимодействующих агентов, или написать вариант модели, начинающейся с одного агента, который клонировал бы себя, оставляя возможность обмена сообщениями у своих клонов.

Второй учебный проект Ball-Communic (<https://scratch.mit.edu/projects/1314960/>) связан с изучением возможностей сбора данных

с искусственного поля. Искусственный мир представляет поле, на котором взаимодействуют две команды и мяч. Каждый из игроков руководствуется следующими простыми правилами:

- 1) поворачивается по направлению к мячу и делает шаг;
- 2) если касается мяча, передает сообщение, что он бьет по мячу.

Агенты-игроки могут различаться по скорости и направленности передвижения. Мяч в этой игре не только реагирует на сообщения игроков, но и записывает всю историю этих сообщений в переменную списка GameStory. Используя детскую среду конструирования цифровых историй, мы создали простое сообщество взаимодействующих агентов. Обратите внимание, что агент-мяч реагирует на сообщения, которые приходят к нему от агентов-котов и записывает в переменную GameStory, кто прислал ему сообщение. В дальнейшем среда Scratch позволяет обрабатывать полученные данные внутри, либо экспортировать и анализировать их при помощи внешних инструментов. Серьезным ограничением для построения и исследования моделей является то, что Scratch может сохранять данные только в виде одномерного списка. Но исходная модель футбольного поля, где действия игроков фиксируются и записываются агентом-мячом, была написана на языке Scratch. Позднее мы существенно расширили и обогатили ее в среде StarLogo Nova.

Snap! расширяет возможности для программирования, которые были у учеников в среде Scratch. В первую очередь это касается возможностей по работе с данными. Одно из ключевых положений дизайна Snap! гласит, что все данные должны быть данными первого класса. Это означает, что все типы данных, включая примитивы (например, числа и строки), списки, спрайты и даже скрипты и блоки, могут быть использованы в качестве входных данных для блоков-функций, возвращены функциями и присвоены переменным. Это позволяет создавать более сложные и интерактивные проекты. Snap! поддерживает гораздо большее разнообразие данных, чем Scratch (рис. 5), а использование списков внутри списков позволяет представлять и конструировать внутри проектов различные структуры данных.

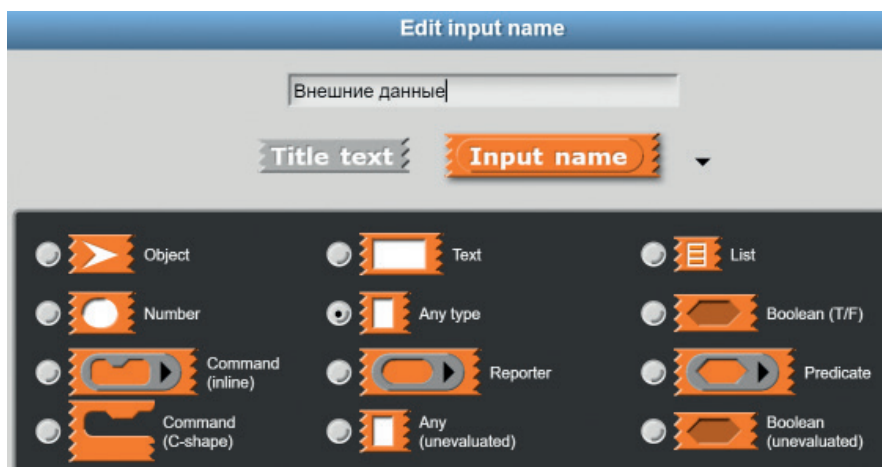


Рис. 5. Разнообразие данных в Snap!

Кроме того, в Snap! есть возможность обратиться с сообщением к конкретному агенту. Например, в версии предыдущего проекта на языке Scratch, где бабочки обменивались сообщениями и все они были получателями всех сообщений, в Snap! есть возможность указать, кто именно из агентов является получателем сообщения.

Для организации учебных курсов по изучению и моделированию сетевых сообществ принципиальное значение имели следующие возможности среды Snap!

1. Возможность использовать внутри среды Snap! готовые наборы данных. Мы можем открывать файлы форматов JSON и CSV, просто помещая их в рабочее пространство Snap! При этом данные автоматически преобразуются в структуру вложенного списка и этому списку присваивается имя переменной. Snap! может обрабатывать как данные внешние звуки и изображения.

2. Возможность обращаться к внешним данным напрямую или через API и преобразовывать эти данные, используя последовательность `split(url)_by`. Последовательность и результат выполнения такого преобразования представлены на рисунке 6.

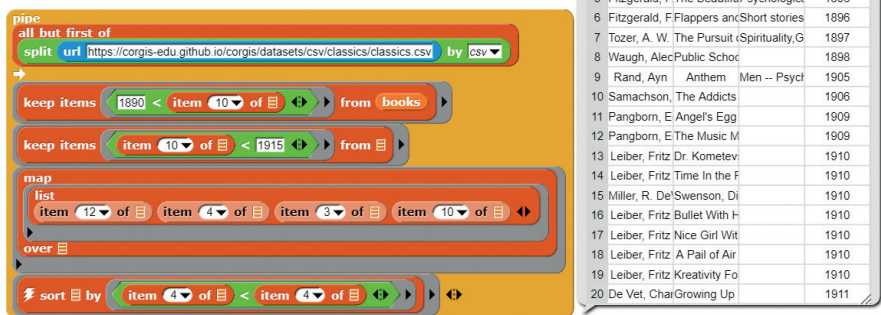


Рис. 6. Выделение, фильтрация и сортировка внешних данных в Snap!

Крайне важно для целей агентного моделирования, что информация о свойствах агентов и их отношениях с другими агентами может обрабатываться внутри самой программы и выдаваться потом в формате доступной для чтения и обработки другими программами. Это обеспечивает связь между Snap!, NetLogo, Python, R, другими средами обработки и визуализации данных.

Многие блоки программирования Snap! совпадают с блоками Scratch, и мы можем представлять и обсуждать их на вики-страницах. Кроме того, специальное расширение позволяет встраивать проекты Snap! в тексты вики-страниц.

StarLogo Nova — среда совместного агентного моделирования в трехмерной среде. Во многом это среда продолжает и развивает традиции языка Scratch. При этом здесь можно ставить серьезные эксперименты и получать воспроизводимые результаты. Центральной метафорой среды является мир (World), в котором можно создавать агентов различных пород.

StarLogo Nova, как и другие среды моделирования, предоставляется бесплатно, работает в браузере и дает возможность учащимся делиться своими проектами и играть в игры, созданные другими участниками. Учителя также могут использовать модели, разработанные сообществом StarLogo Nova, для объяснения

сложных понятий. Пользователи могут создавать простые или масштабные модели, используя существующую библиотеку агентов или импортируя свои звуки и 3D-модели в формате Collada. Эта среда активно используется в естественно-научном образовании. В качестве примера следует привести проект GUTS, в котором взаимосвязаны образовательные стандарты вычислительного мышления и исследовательской деятельности.

При создании проекта система предлагает выбор из трех вариантов: 1) начать проект с чистого листа, 2) начать создавать игру, в которой мир будет показан с точки зрения главного персонажа, 3) создать модель биологического процесса. Если выбрать игру, то система показывает мир с точки зрения геометрической фигуры, вокруг которой двигаются другие фигуры. Существует множество развитий этого сюжета, когда в систему добавляются правила взаимодействия героя с другими агентами. Если выбрать биологическую модель, то система изначально создает не только взаимодействующие породы, но и график, на котором будет отображаться численность здоровых, инфицированных и поправившихся экземпляров породы.

Мир StarLogo наследует основные понятия Scratch. Здесь есть:

- рабочая область экрана, где агенты совершают свои действия — зелёная область в центре экрана;
- вокруг зеленой области, где действуют агенты, расположено черное поле, где автор может размещать управляющие кнопки и информационные виджеты;
- область хранения команд, которые собраны в разноцветных ящиках;
- страницы отдельных пород, на которых из разноцветных команд собираются процедуры, управляющие поведением агентов данной породы.

Вкладка управления проектами расположена сверху экрана и здесь же находится кнопка запуска кода Run Code. Не забывайте запускать код всякий раз, когда вы вносите изменения в свой проект. Без этой кнопки никаких изменений не происходит. Следующая важная кнопка, с которой начинаются события в мире,

называется Setup. Обычно эту кнопку использует Мир / The World при обновлении и запуске проекта. На рисунке 7 представлена последовательность действий Мира, когда нажата кнопка Setup.

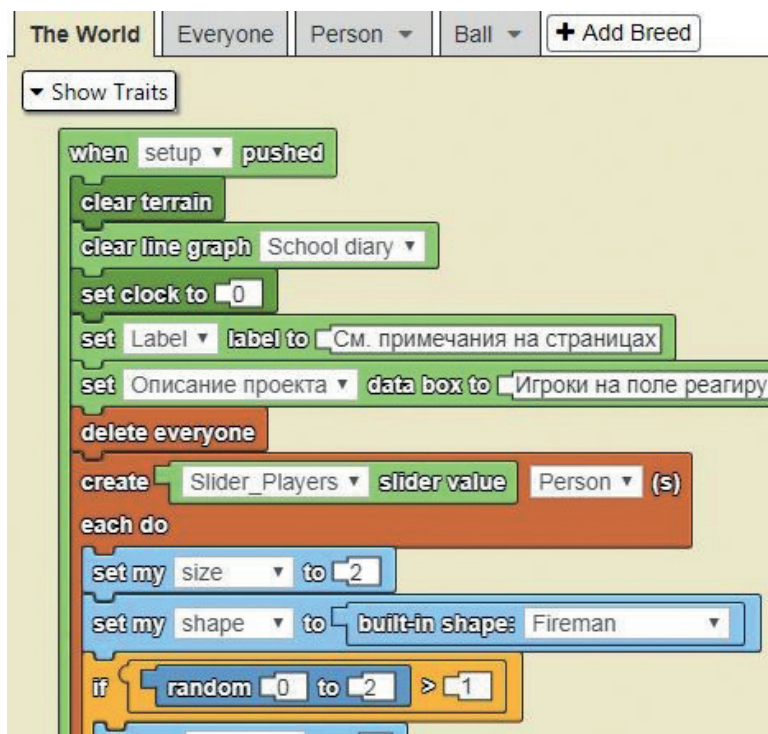


Рис. 7. Запуск Мира и пород

Перевод:

Когда нажата кнопка Setup

Очистить местность

Очистить граф (таблицу) журнала

Установить время в 0

Установить текст «...» в описание проекта

Удалить всех агентов на местности

Создать агентов породы Person по числу на рычажке и пусть каждый из них

Станет размером 2

Станет по своей форме как Fireman

Важно, что Мир знает свойства всех существующих пород и может обращаться к агентам любых из них. Кроме взаимодействия на основании информации о положении и свойствах других агентов в StatLogo Nova есть механизм столкновений (collision), когда агенты сталкиваются друг с другом на поле и дальнейшее развитие сюжета зависит от пород и свойств тех, кто столкнулся.

Команды управления взаимодействия агентов и определения агентов, обладающих определенными свойствами, представлены на рисунке 8.

	Если столкнулся с агентом породы [], то делай следующее: collidee — тот, с кем столкнулся. Например, delete collidee Число агентов породы [] на расстоянии [] Число агентов породы [] на расстоянии [] с определенным свойством (например, Team 1) Ближайший агент породы [] на расстоянии [] Число агентов породы [] на расстоянии [] с определенным свойством
--	---

Рис. 8. Взаимодействия

При помощи этих команд мы можем задать условия программы, управляющие перемещением агентов, можем научить агентов реагировать на свойства поля, по которому они перемещаются. Мы можем определить действия, которые будут совершаться в системе при столкновении агентов. Такие столкновения в системе называются коллизиями. Если мы не указываем агентам, как действовать в случае столкновений, то они будут просто проходить через друг друга. Если же мы воспользуемся блоком коллизий, то мы сможем программировать события, которые будут происходить при столкновениях агентов данной породы с агентами этой же или других пород. Например, на следующем примере показана модель распространения инфекции, когда синие здоровые черепахи сталкиваются с больными красными и в результате столкновения меняют свой цвет. В следующем блоке команд (рис. 9) черепаха при столкновении проверяет цвет агента, с которым она столкнулась, и если этот цвет красный, то она и сама окрашивается в красный цвет.

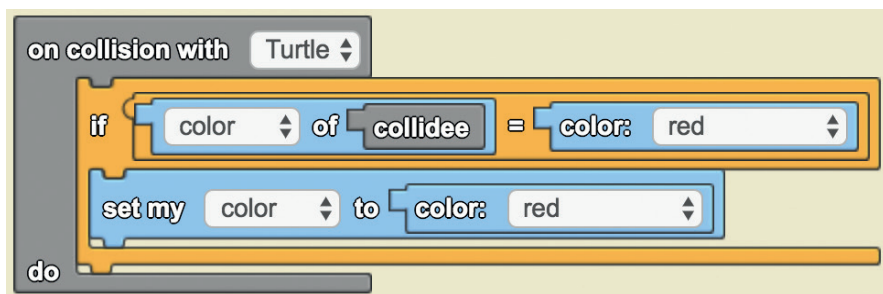


Рис. 9. Выбор действий при столкновении

Начать знакомство с возможностями StarLogo TNG лучше всего с просмотра примеров проектов, созданных в этой среде. В рамках учебных курсов, связанных с использованием возможностей StarLogo TNG мы разбираем проект Football Game with a Log File of Activities, доступный в сети по адресу: <https://www.slnova.org/patarakin/projects/691186/>

Мы продолжаем моделировать отношения агентов — игроков и мячей — на футбольном поле в более богатой возможностями

по сравнению с Scratch среде. Интерфейс для управления свойствами модели представлен на рисунке 10. Пользователь может управлять числом игроков на поле и расстоянием, на котором игроки видят мяч — виджет Vision. На экране мы видим зеленое поле, где взаимодействуют игроки, и таблицу, в которой записываются действия — ID игрока и ID мяча, по которому он ударяет.

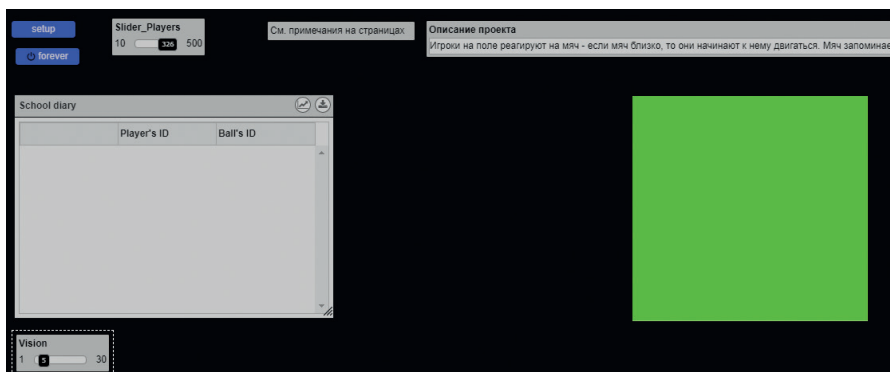


Рис. 10. Интерфейс футбольного поля

В данном примере мы рассмотрим историю, когда объект «мячик» медленно катится по полю и реагирует на присутствие игроков, которые находятся поблизости. Внешне это будет выглядеть так, как будто игроки бегут к мячу и наносят по нему удар. А в ответ на этот удар мячик отлетает в сторону направления удара. А еще (и это для нас самое интересное), мяч будет собирать данные о том, какой игрок, из какой команды, в каком месте и в каком направлении нанес удар. В StarLogo Nova мы создаем и управляем не отдельными экземплярами агентов, а породами (breed). Следовательно, на странице породы записываются правила, которыми будут руководствоваться все агенты этой породы. Например, все игроки на поле будут следовать правилам, которые описаны на странице Person.

Обратим внимание на первое правило на странице: «Пока кнопка Forever включена, вызывай процедуру FB_2». А дальше мы посмотрим на команды, которые мы можем использовать.

У каждой породы есть обязательные свойства, которые должны быть у каждого агента — это форма, цвет и размер. Кроме того, мы можем добавлять породе любые свойства, которые покажутся нам нужными для работы модели. Например, мы можем добавить породе Person свойство Team — и теперь игроки на местности нашего Мира могут относиться к разным командам.

Команды управления свойствами агента собраны на голубой вкладке Traits. Здесь мы можем устанавливать как обязательные, так и дополнительные свойства каждой породы агентов.

Самый простой случай — Set my Size to (5) — установить размер данного агента.

Если мы хотим установить цвет, то обращаемся к Set my Color to [color (выбор цвета)].

При выборе формы для агента можно использовать встроенную библиотеку со множеством форм, а также загружать в систему самостоятельно созданные формы. Команды перемещения агентов почти не отличаются от обычных команд к Черепашке или котенку в Scratch: вперед, назад, налево, направо. Поскольку среда трехмерная, то есть еще команды «вверх» и «вниз» и команда телепортации в трехмерную точку $x : y : z$.

Взаимодействие агентов может быть основано на том, что у них есть данные о положении других агентов около себя. Например, в следующей процедуре, если наш агент чувствует, что на расстоянии двух шагов от него число игроков больше 0, то он вызывает процедуру записи в журнал и передает ей в качестве параметра информацию о ближайшем игроке на расстоянии двух шагов. Процедура создается из множества блоков команд. Чтобы использовать процедуру, ее необходимо вызвать через команду Call (имя процедуры). Скрипты, управляющие поведением агентов породы футболистов, представлены на рисунке 11.

В верхней части рисунка мы видим свойства, которыми обладают агенты данной породы. Здесь пока перечислены только их форма, цвет и размер. Но, используя кнопку Add Custom Trait, мы можем добавлять игрокам новые свойства. Например, мы можем относить их к разным командам, регулировать силу

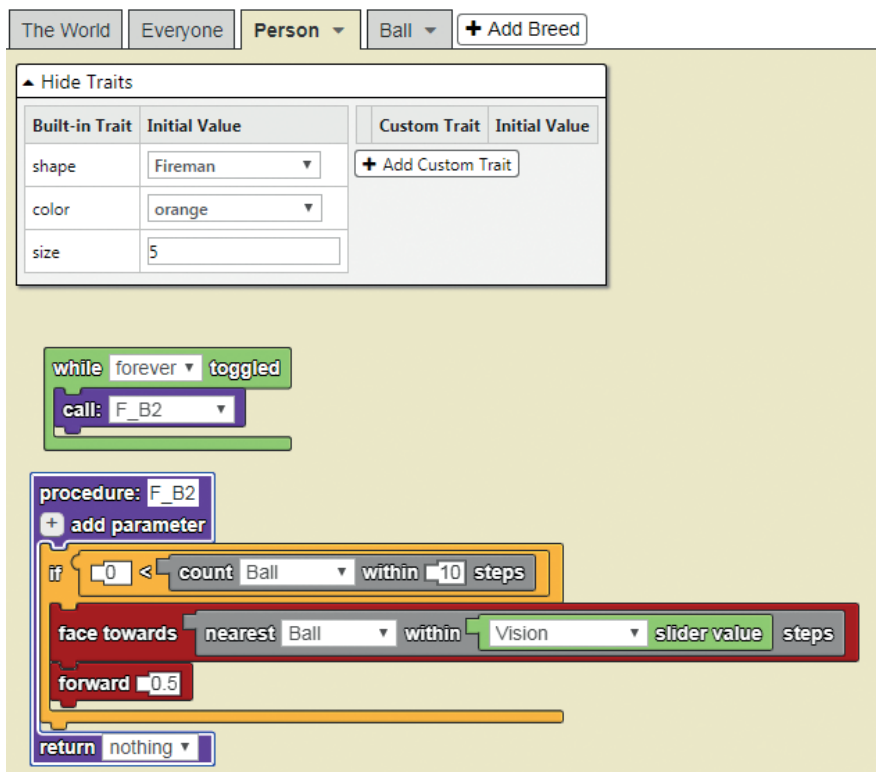


Рис. 11. Управление поведением игроков

и точность удара, указывать части поля, на которых они предпочитают играть.

Первая порода — для футболистов. Агенты этой породы в своем поведении руководствуются следующим правилом: *если рядом с тобой находится мяч (если ты его видишь), то повернись в направлении этого мяча и сделай шаг вперед.*

Вторая порода создана для футбольных мячей. Эти агенты следуют следующим правилам: *если расстояние до футболиста меньше двух шагов, то запиши в лог-файл ID-номер агента и свой собственный ID и переместись на десять шагов от агента.*

Процедура, управляющая поведением мяча, представлена на рисунке 12. Согласно этой процедуре, если рядом с мячом находится

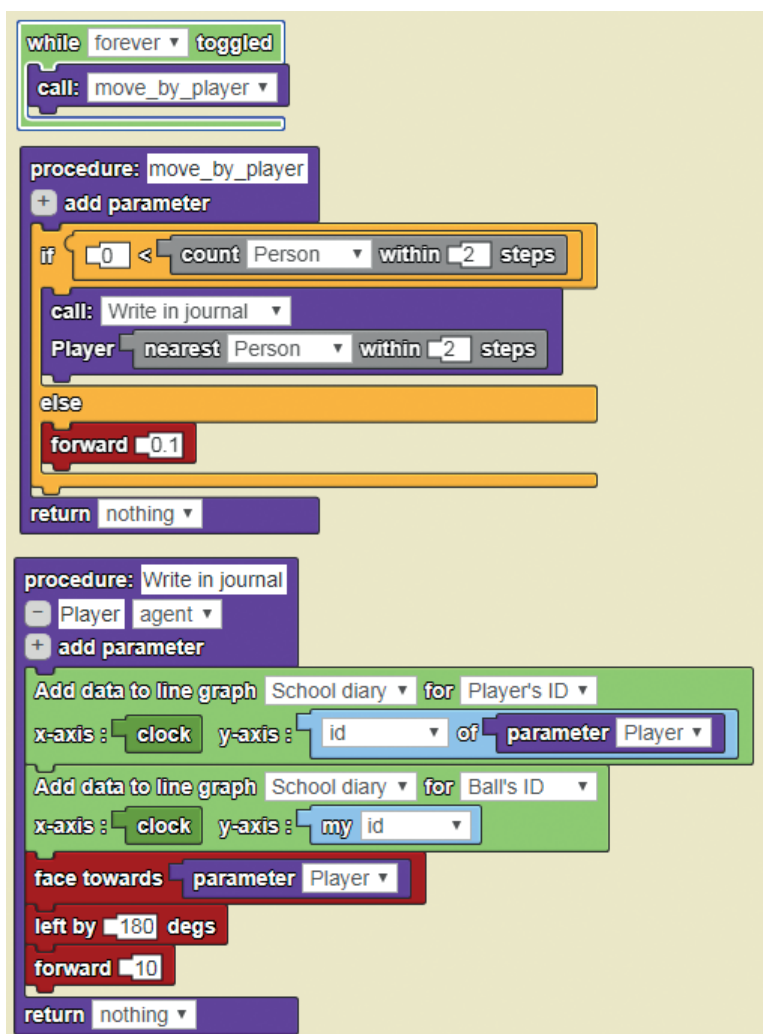


Рис. 12. Управление поведением мячей

агент(ы) из породы игроков, то мяч записывает в журнал ID ближайшего игрока и свой собственный ID. После этого мяч отлетает от игрока. Мы можем считывать из системы не только ID игрока, но и другие его свойства и увязывать поведение мяча не только с тем направлением, куда смотрит игрок, но и с другими свойствами агента игрока.

Следующая процедура будет записывать в журнал данные о тех игроках, которых мяч встретил на футбольном поле. Процедура запускается самим мячом, когда он чувствует, что рядом с ним кто-то есть. В этом случае мяч записывает в соответствующие ячейки таблицы данные об игроке (рис. 13).

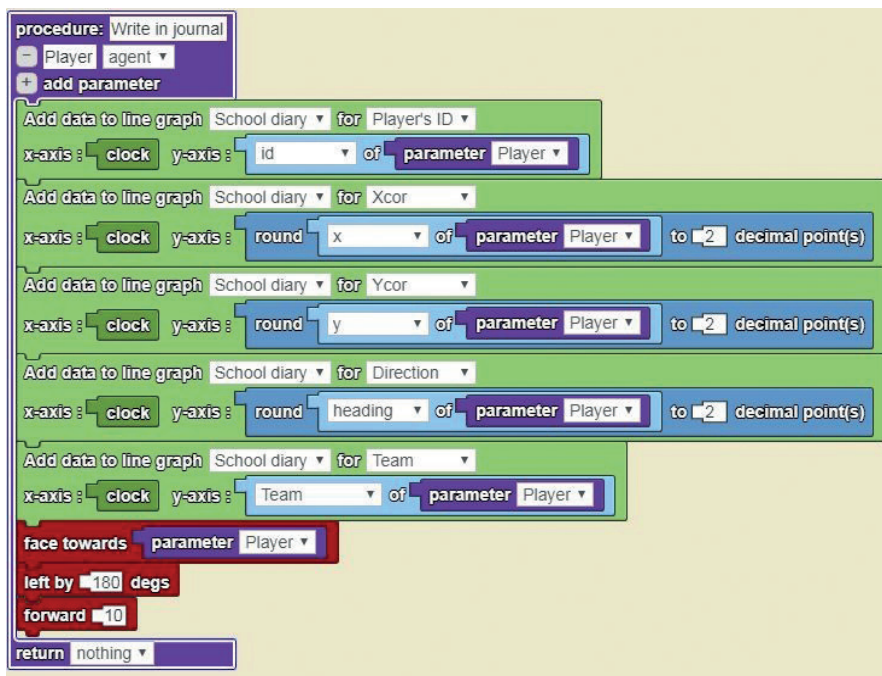


Рис. 13. Запись данных в таблицу

Линейный граф всегда может быть преобразован в таблицы или сохранен в виде CSV-файла. Перед тем как запустить модель, имитирующую поведение множества агентов, пользователь может выбрать число игроков и число мячей, которые будут участвовать в игре. В представленном на рисунке 14 примере на искусственном игровом поле взаимодействовали 300 игроков и 7 мячей. После нескольких минут игры запись действий была сохранена в CSV-формате, перенесена в среду R и конвертирована в сетевую диаграмму. Ребра диаграммы соединяют между собой игроков

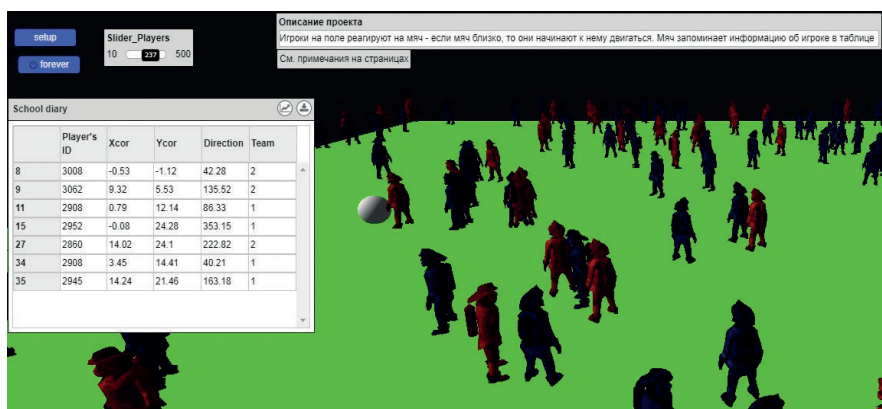


Рис. 14. Данные на экране

и мячи, которых они касались в ходе игры. Даже такая простая модель позволяет нам получить и исследовать различные типы сетевых структур в зависимости от того, как далеко участники видят мяч и насколько далеко мяч перемещается после удара.

NetLogo — это бесплатная мультиагентная программируемая среда моделирования с открытым исходным кодом. По данным сайта NetLogo используют тысячи учеников, преподавателей и исследователей во всем мире. NetLogo написан на Java и Scala. Язык NetLogo достаточно прост, и ученики, и учителя могут создавать в этой среде свои собственные учебные модели. В то же время NetLogo — это достаточно мощный язык для построения исследовательских моделей и проведения исследований в области сложных адаптивных систем. NetLogo удобно использовать для моделирования сложных, развивающихся во времени систем. Создатель модели может давать указания сотням и тысячам независимых агентов, действующих параллельно. Это открывает возможность для объяснения и понимания связей между поведением отдельных индивидуумов и явлениями, которые происходят на макроуровне.

У черепашек Netlogo есть свойства: номер, цвет, координаты и т. д. Кроме того, мы можем задавать черепашкам новые свойства — `turtles-own`. Свойства всегда задаются в начале программы. Например, зададим для черепашек свойство `Group`.

Свойство будет иметь два значения — ложь или истина. Принадлежит черепашка к группе или не принадлежит.

Внутри `turtles-own []` можно перечислить множество свойств, которыми будут обладать черепашки. Если мы создали новую породу черепах, то с этой породой точно так же можно создать ее свойства.

Создадим первую процедуру, которая будет очищать все имеющиеся объекты и значения. Далее следует программа с комментариями. Комментарии задаются двумя символами — `;;`;

Создаем новых черепах. Черепахи создаются по порядку и поворачиваются последовательно. Когда мы им потом даем команду разойтись, они образуют круг:

```
to setup
  ca ; — очищаем значения
  ;; каждая черепашка располагается в случайной точке
  ;; экрана
  crt 50 [ ;; создаем 50 черепах и разбрасываем их
    по экрану
    setxy random-pxcor random-ycor
    set group false
  ]
  cro number_of_turtles [fd max-pxcor - 2]
end
```

Этого же результата можно достичь, если мы используем команду `layout-circle`:

```
to circle
  layout-circle turtles max-pxcor - 2
end
```

Мы умеем создавать черепашек и расставлять их по экрану. Теперь попробуем собрать черепашек одинакового цвета. Группируем черепах по цветам. Первая версия: черепашки объединяются, но делают много лишних движений:

```
to collect_color
  ask other turtles with [color = [color] of myself]
  [face myself jump (distance myself) - 1]
end
```

Во второй версии задаем черепашкам свойство принадлежности к группе и в начале устанавливаем значение этого свойства в false:

```
if group = false [ask other turtles with [(color =
[color] of myself) and (group = false)]
[
; черепашка прыгает в сторону вызывающего агента
face myself jump (distance myself) - 1
; черепашка устанавливает свою принадлежность к группе
в true - и теперь ее не будут вызывать
set group true create-link-to myself]]
end
```

Пятна (patches) — это агенты NetLogo, которые связаны с конкретными координатами на экране. К пятну или точке всегда можно обратиться, указав его координаты. Например:

```
ask patch 0 0 [set pcolor yellow]
```

Для пятен существуют ограничения. Например, невозможно велеть пятну умереть и создать новое пятно. Однако пятно может родить новую черепашку (sprout) или черепашку какой-то специальной породы (sprout-breed). По команде sprout в данной точке появляется черепашка.

Ключевые команды для пятен:

```
clear-patches (cp) ;; Очистить, стереть все пятна
и вернуть их в исходное значение - черный цвет
diffuse patch-variable number ;; значение переменной
раздается на ближайших соседей - диффузия цвета, запаха
на поверхности
diffuse4 patch-variable number ;; значение переменной
раздается на 4-х соседей - ближайшие соседние точки
по прямой
import-pcolors ;; считать цвета из файла - картинки
patch-at dx dy ;; пятно с координатами относительно
данного агента
```

Например,

```
ask patch 2 5 [set pcolor yellow ask patch-at 2 2 [set
pcolor red]]
```

```
patch-left-and-ahead angle distance ;; пятно слева
и сверху
patch-at-heading-and-distance ;; пятно по направлению
и на расстоянии
ask patch-at-heading-and-distance -90 1 [ set pcolor
green ]
```

Примеры команд и процедур с пятнами.

✓ Окрасить все пятна в зависимости от расстояния до центра:

```
ask patches [set pcolor distanceexy 0 0]
```

✓ Покрасить точки вокруг центральной точки в зеленый цвет:

```
ask patch 0 0 [set pcolor yellow ]
ask patch 0 0 [set pcolor yellow ask neighbors4 [ set
pcolor green ]]
ask patch 0 0 [set pcolor yellow ask neighbors4 [ set
pcolor green ask neighbors [set pcolor red] ]]
```

и так далее, если хотим перекрашивать соседей.

Связи в NetLogo такие же агенты, как черепахи NetLogo и пятна NetLogo. Две черепахи всегда связаны между собой (два узла). Если одна из черепах погибает, то погибает и связь. Связи бывают направленными и ненаправленными.

Ненаправленные связи создаются командами к черепахам: `create-link-with` или `create-links-with`. Например:

```
ask turtle 0 [create-link-with turtle 1] ;
; - команда черепахе 0 создать связь с черепахой 1
ask turtle 0 [create-links-with other turtles]
;; команда черепахе 0 создать связи со всеми другими
черепахами
ask turtles [create-links-with other turtles]
;; команда всем черепахам создать связи со всеми другими
черепахами
```

Направленные связи:

`create-links-from` или `create-link-from` направленная связь других агентов к данному агенту.

Например:

```
ask turtle 0 [create-link-from other turtles]
```

create-link-to или create-links-to направленная связь от данного агента к другому или другим
ask turtle 0 [create-link-to other turtles]

Связи между породами.

Породы NetLogo связываются такими же командами, как и черепахи: create-<breed>-from, create-<breeds>-from, create-<breed>-to, create-<breeds>-to, create-<breed>-with, create-<breeds>-with

В NetLogo есть команды, которые указывают агентам, что делать: переместиться, создать черепашку, установить связь и т. д. Обычно эти команды начинаются с глаголов die, jump, inspect.

Процедуры можно создавать: в этом случае процедура начинается со слов «То Имя_Процедуры» и заканчивается словом End.

Кроме того, есть процедуры, которые получают на входе данные, обрабатывают эти данные и выдают результат. Эти процедуры обязательно начинаются со слова to-report и содержат команду вывода значения — report.

```
to-report absolute-value [number]
  ifelse number >= 0
    [ report number ]
    [ report (- number) ]
end
```

В этом примере процедура absolute-value получает число и возвращает его абсолютное значение.

Начиная с версии 6.0 в NetLogo появились анонимные процедуры. Например:

```
to logs_to_sociogram
  ask patches [set pcolor 0]
  foreach edits [ [?1] -> ;; это вызов анонимной процедуры
    let friend1 item 0 ?1
    let p1 item 1 ?1
    let friend2 first first filter [ [??1] -> (p1 = item
      1 ??1) and ("create" = item 2 ??1) ] wikilog
    if friend1 != friend2 [
```

```

ask turtle friend1 [ create-friendbond-to turtle
friend2 ]
] ]
repeat 2 [layout-spring turtles links 1 10 12 ]
end

```

В рамках данной главы, где мы сравниваем возможности разных сред программирования внутри учебных курсов по изучению совместной сетевой деятельности и сетевых сообществ, необходимо обратить внимание, что веб-интерфейс программы NetLogo Web содержит те же возможности, которые содержатся в средах Scratch, Snap!, StarLogo Nova. Особенно заметна близость StarLogo Nova и NetLogo Web, если мы откроем вкладку NetLogo Code и перейдем в режим редактирования пользовательского интерфейса (см. рис. 15). Мы можем добавлять на экран новые интерактивные элементы (кнопки, рычажки, графики и т. п.) и вносить изменение в код программы.

На рисунке 15 представлена модель формирования стай. Эта модель очень часто используется при обсуждении самоорганизации агентов на основании простых правил. Мы рассмотрим процедуры, управляющие поведением отдельных агентов, сравнивая возможности среды Snap! и среды NetLogo.

Модель формирования стай на языке Snap! доступна на сайте Snap!: https://snap.berkeley.edu/project?username=patarakin&projectname=Flocking_Snap

Модель формирования стай на языке NetLogo доступна в библиотеке учебных проектов: <https://ccl.northwestern.edu/netlogo/models/Flocking> — и на сайте NetLogo Web.

В качестве объектов управления и анализа в этой модели используются подвижные агенты. Исторически сложилось так, что все подвижные агенты-исполнители в логоподобных средах называются черепахами. При изучении конкретных экологических или социальных феноменов черепахи могут надевать на себя маски птиц, волков, овец, светлячков, учеников или учителей.

Эта модель — попытка имитировать процесс формирования стаи птиц в отсутствие каких-либо лидеров. Каждая птица следует общему набору правил и из этого общего поведения и формируется

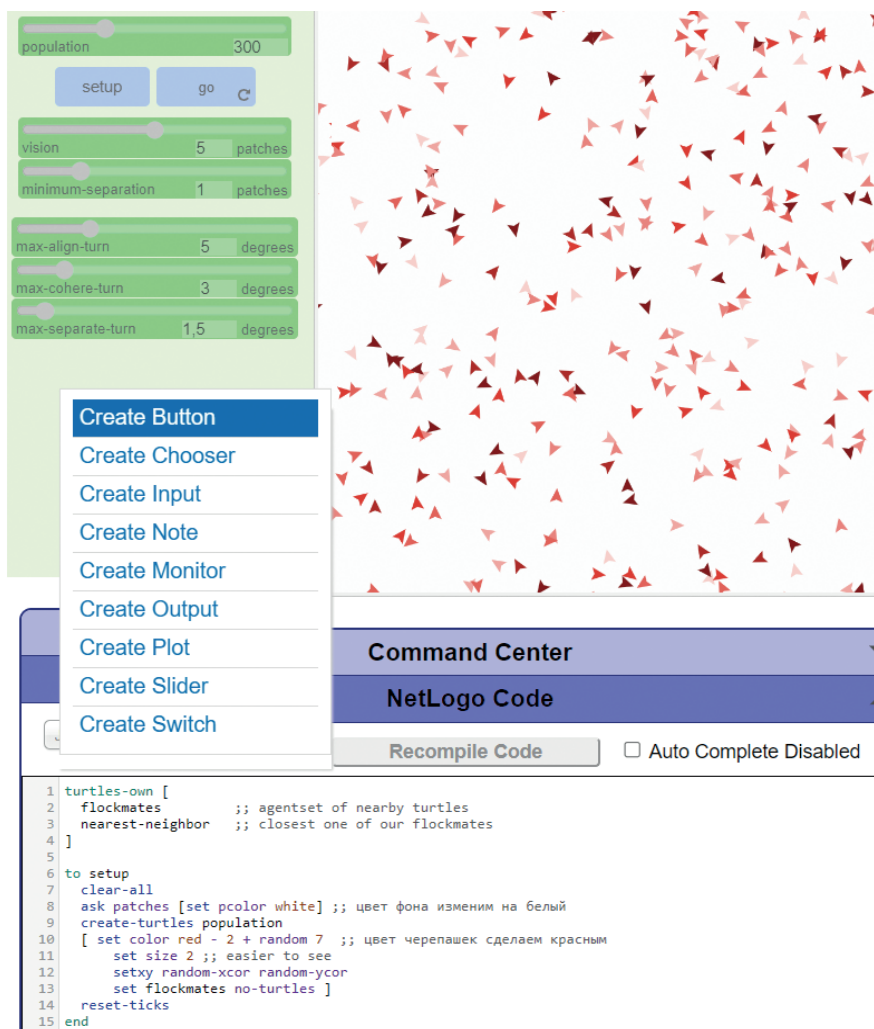


Рис. 15. Редактирование модели NetLogo Web

стая. Стаи птиц можно рассматривать как пример возникновения стайных свойств в результате того, что отдельные животные двигаются в ответ на движения других животных. Как сказано в описании модели, все птицы следуют трем правилам: выравнивание, разделение и сплоченность. На уровне действий эти правила на языке NetLogo записаны как процедуры к выполнению:

```

to flock
  find-flockmates
  if any? flockmates
    [ find-nearest-neighbor
      ifelse distance nearest-neighbor < minimum-separation
        [ separate ]
        [ align
          cohere ]
    ]
  end
end

```

Перечень практически тех же команд по отношению к клону черепахи в среде Snap! представлен на рисунке 16.

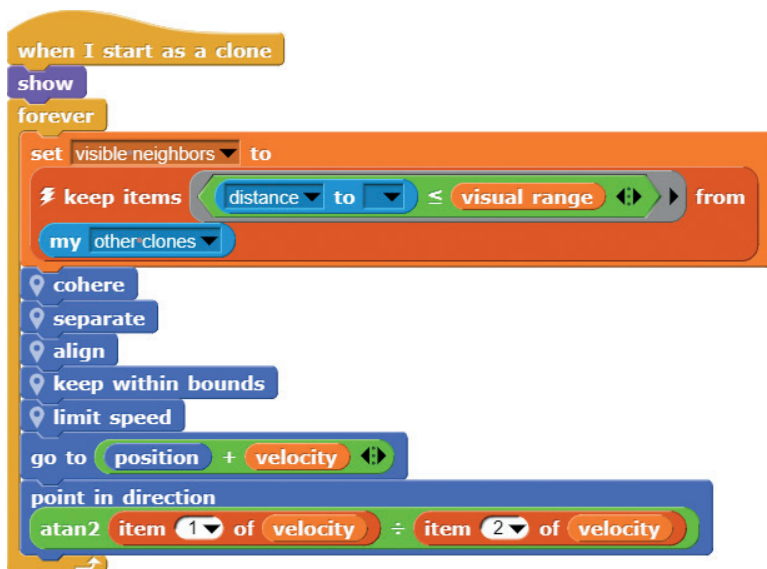


Рис. 16. Блоки управления поведением клона

Разделение означает, что птица поворачивается, чтобы избежать столкновения с другой птицей, которая летит слишком близко.

```

to separate
  turn-away ([heading] of nearest-neighbor) max-separa-
  te-turn
end

```

Командный блок разделения на языке Snap! представлен на рисунке 17.

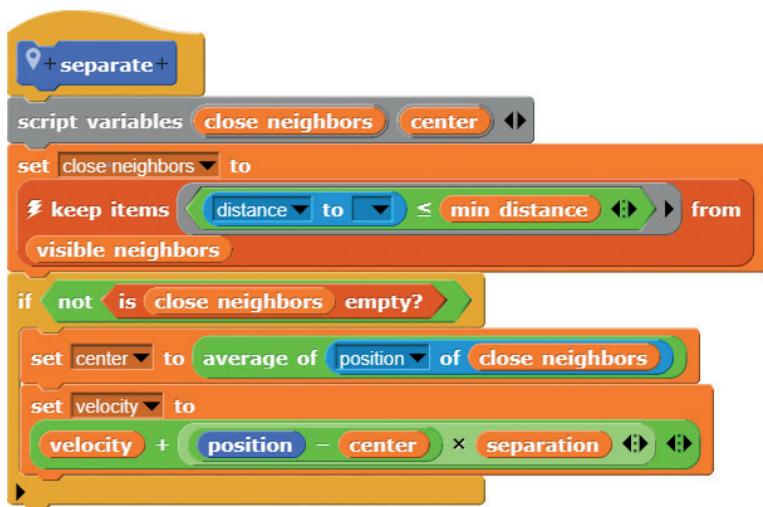


Рис. 17. Блок разделения

Выравнивание означает, что птица поворачивается так, чтобы двигаться в том же направлении, что и ближайшие птицы.

```
to align ;; turtle procedure
  turn-towards average-flockmate-heading max-align-turn
end
```

Командный блок выравнивания на языке Snap! представлен на рисунке 18.

Сплоченность означает, что птица будет двигаться к другим ближайшим птицам.

```
to cohere ;; turtle procedure
  turn-towards average-heading-towards-flockmates
max-cohere-turn
end
```

Командный блок сплоченности на языке Snap! представлен на рисунке 19.

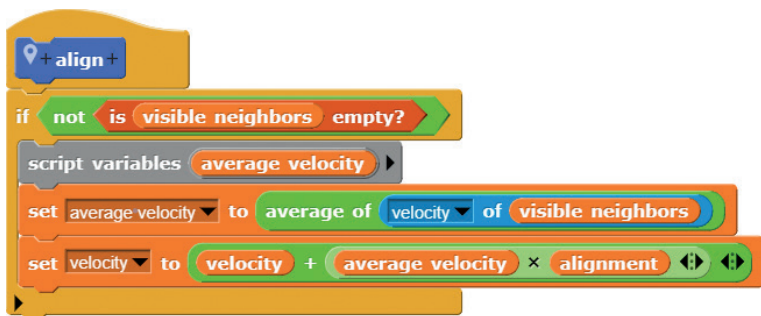


Рис. 18. Блок выравнивания

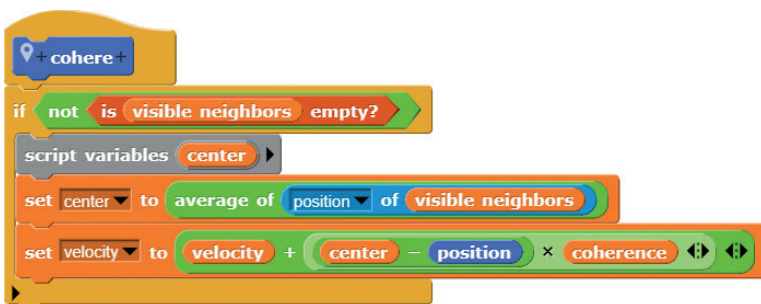


Рис. 19. Блок сплоченности

Когда две птицы находятся слишком близко, правило разделения отменяет действие двух других.

Правила «выравнивание» и «сплоченность» не начнут действовать до тех пор, пока не будет достигнуто минимальное разделение. Все три правила влияют только на направление птицы и никак не влияют на скорость полета.

Отдельного внимания заслуживает встроенный в среду NetLogo инструмент BehaviorSpace — программный инструмент, интегрированный с NetLogo, который позволяет проводить эксперименты с моделями. BehaviorSpace запускает модель много раз, систематически изменяя настройки модели и записывая результаты каждого запуска модели, что позволяет исследовать «пространство» модели возможного поведения и определять, какие комбинации настроек вызывают интересующее поведение. У многоагентных моделей

часто есть много настроек, каждая из которых может принимать различные значения. Вместе они образуют пространство параметров модели, размеры которого представляют собой количество настроек, а каждая точка представляет собой определенную комбинацию значений. Запуск модели с разными настройками (а иногда даже с одинаковыми) может привести к совершенно разному поведению моделируемой системы. Например, если модель имеет четыре ползунка — число, длину цикла, длину вспышки и количество миганий до сброса, — которые имеют приблизительно 2000, 100, 10 и 3 возможных значения соответственно, то существует 6 000 000 возможных комбинаций значений ползунка. BehaviorSpace предлагает гораздо лучший способ решить эту проблему. Если вы укажете подмножество значений из диапазонов каждого ползунка, то инструмент будет запускать модель с каждой возможной комбинацией этих значений и во время каждого запуска модели записывать результаты. После завершения всех прогонов создается набор данных, который можно открыть в другом инструменте, таком как электронная таблица, база данных или приложение для научной визуализации, и изучить его. В своей исследовательской и педагогической практике мы использовали последовательность инструментов BehaviorSpace NetLogo -> язык R. Необходимо отметить, что язык R в настоящее время является одним из главных инструментов исследования и анализа данных и все работы по анализу и визуализации данных в данной монографии выполнены с использованием этого языка. Современные версии языка (Wickham, Grolemund, 2016) гораздо проще для освоения. На базе R создаются разнообразные веб-приложения подобные JASP (<https://jasp-stats.org/>). Можно утверждать, что среда R в настоящий момент уже может рассматриваться как дружелюбная для преподавателей, студентов и школьников лаборатория для статистического анализа.

Позволяя исследовать все «пространство» поведения, которое может демонстрировать модель, BehaviorSpace является мощным помощником разработчика моделей. При запуске инструмента BehaviorSpace появляется окно, в котором пользователь указывает для программы задания к выполнению:

- Как назвать эксперимент?
- Какие сценарии эксперимента нужно создать и какие значения в этих сценариях будут принимать глобальные переменные?
- Сколько раз нужно повторить исполнение каждого сценария?
- В какой последовательности нужно проводить выполнения?
- Какие данные нужно собирать в ходе выполнения сценария или после его завершения?
- Какой параметр использовать для остановки сценария, если сценарий не завершается автоматически?

Начало работы с настройкой инструмента BehaviorSpace представлено на рисунке 20. Мы выбираем инструмент в верхнем меню на вкладке Tools.

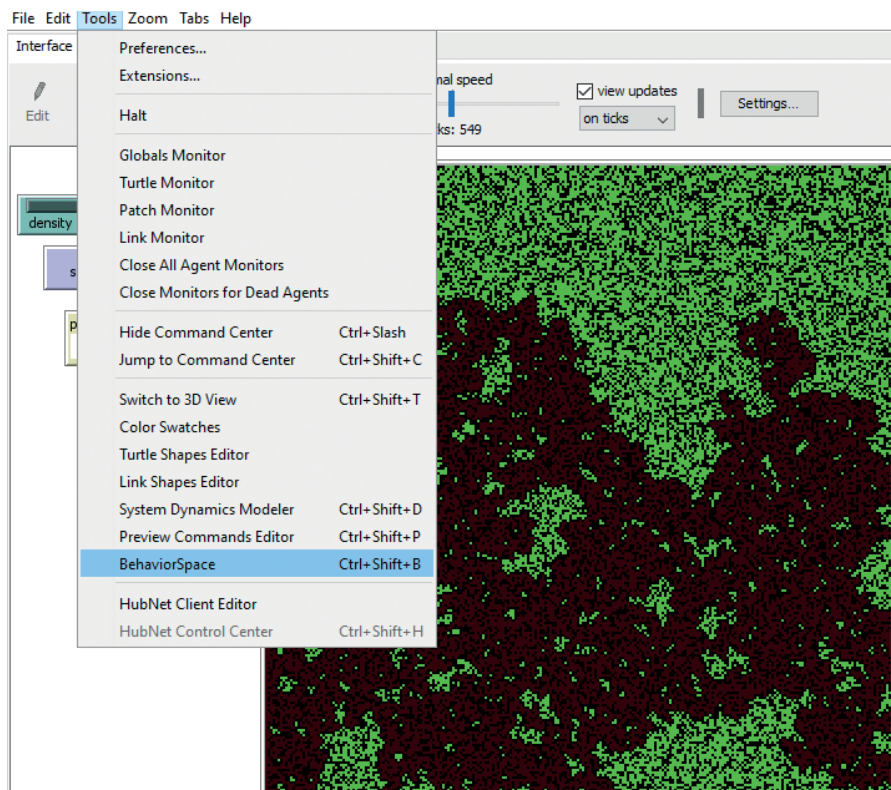


Рис. 20. BehaviorSpace NetLogo

Например, мы укажем для эксперимента, что необходимо изменять значение переменной `density` ["density" [51 0.5 70]], начав со значения 51 и увеличивая плотность на 0,5 в каждой серии. При этом мы укажем, что каждая серия должна повторяться десять раз. И каждый раз в конце эксперимента мы будем измерять процент сгоревших деревьев. В данном случае нам не нужно указывать условие завершения программы, поскольку мы знаем, что каждый прогон эксперимента обязательно завершается. Экран с условиями эксперимента представлен на рисунке 21.

Таким образом мы задали 39 серий экспериментов, каждая из которых будет повторяться десять раз. При запуске эксперимента программа предлагает нам выбрать формат, в котором будут сохраняться результаты. В данном случае мы выбираем Table output (см. рис. 22).

Результаты экспериментов записываются программой в таблицу CSV: в данном случае это был файл `Fire_2_table.csv`. Мы можем открыть этот файл в любом текстовом редакторе, приложениях для работы с электронными таблицами или загрузить в среду для статистической обработки языка R. Покажем, как выглядят первые несколько строк в текстовом редакторе:

```
"BehaviorSpace results (NetLogo 6.2.0)"
"Fire_exp.nlogo"
"experiment"
"05/06/2021 18:19:12:697 +0300"
"min-pxcor", "max-pxcor", "min-pycor", "max-pycor"
"-125", "125", "-125", "125"
"[run number]", "density", "[step]", "(burned-trees /
initial-trees) * 100"
"2", "51", "41", "2.036468450404161"
"4", "51", "82", "3.3096705897004792"
```

Обратите внимание, что первые шесть строк в таблице содержат описание условий эксперимента: в какой среде он проводился, как называлась модель, какими были размеры искусственного мира. При анализе и визуализации данных эта информация не используется, поэтому при загрузке файла в R мы указываем число строк, которые нужно пропустить: `skip = 6`:

```
df <- read_csv("Fire_2_table.csv", skip = 6)
```

 Experiment
 ✕

Experiment name

Vary variables as follows (note brackets and quotation marks):

['density' [51 0.5 70]]

^

 v

Either list values to use, for example:
 ["my-slider" 1 2 7 8]
 or specify start, increment, and end, for example:
 ["my-slider" [0 1 10]] (note additional brackets)
 to go from 0, 1 at a time, to 10.
 You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions

run each combination this many times

☒ Run combinations in sequential order

For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
 sequential order: 1, 1, 2, 2, 3, 3
 alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

(burned-trees / initial-trees) * 100

^

 v

one reporter per line; you may not split a reporter
 across multiple lines

☐ Measure runs at every step

if unchecked, runs are measured only when they are over

Setup commands:

setup

^

 v

Go commands:

go

^

 v

☐ Stop condition:

the run stops if this reporter becomes true

☐ Final commands:

run at the end of each run

Time limit

stop after this many steps (0 = no limit)

OK

Cancel

Рис. 21. Условия эксперимента

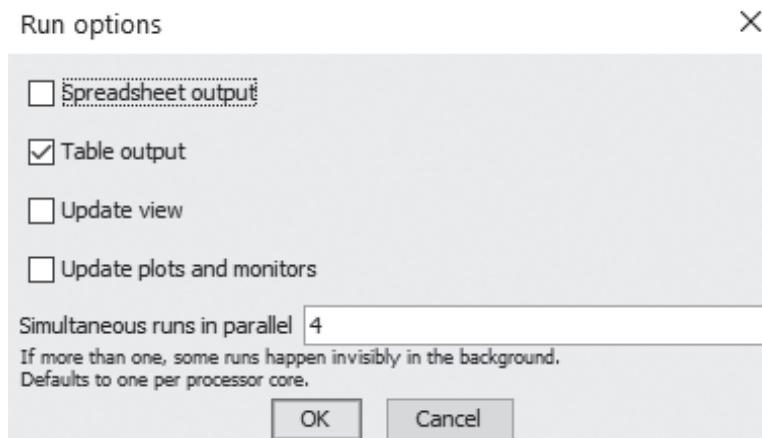


Рис. 22. Выбор формата записи

Загрузив данные, мы тут же можем представить на графике Box Plot зависимость процента сгоревших деревьев от плотности леса:

```
boxplot(persent_burned ~ density, data=df, main="Данные  
о пожарах", xlab="Плотность", ylab="Процент сгоревших")
```

На рисунке 23 мы видим, что максимальный межквартильный размах и вариативность процента сгоревших деревьев наблюдается на промежутке плотности леса 57–60 %.

Модель помогает исследовать то, как плотность деревьев в лесу влияет на распространение пожаров. При этом модель знакомит нас не столько с лесными пожарами, сколько с нелинейным поведением сложных систем, когда очень небольшое изменение в значении данных может привести к очень большим изменениям в результатах. При переходе границы плотности леса от 57 к 60 % мы наблюдаем резкое снижение устойчивости системы к пожару.

Приведем еще один пример использования инструмента BehaviorSpace для сбора данных эксперимента из модели с разными скоростями размножения. Модель доступна в библиотеке учебных проектов: <https://ccl.northwestern.edu/netlogo/models/SimpleBirthRates> — и на сайте NetLogo Web.

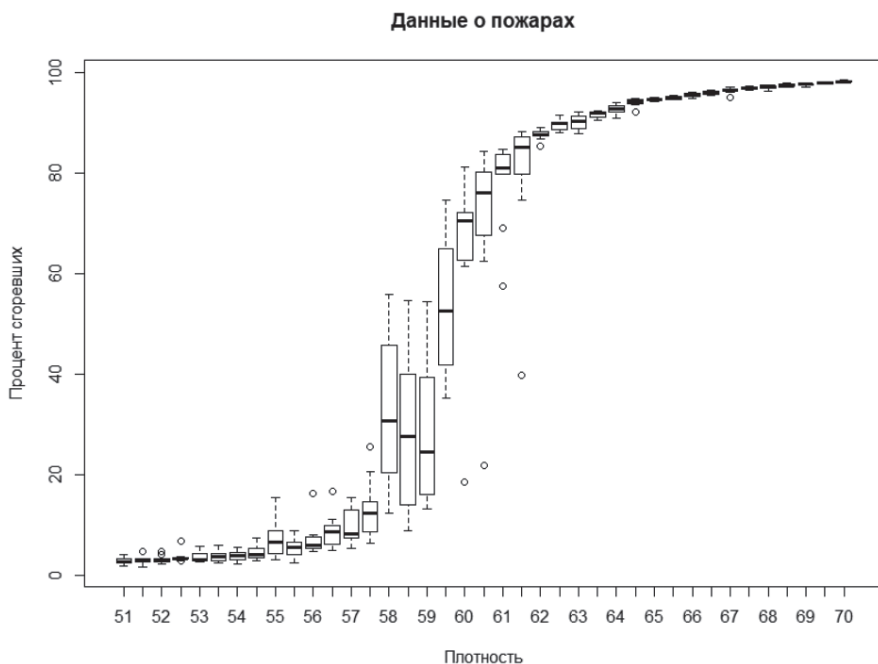


Рис. 23. Box Plot данных

Это простая модель популяционной генетики, когда на одной территории есть две популяции: красные и синие. У каждой популяции есть настраиваемый уровень рождаемости. Красные и синие черепахи перемещаются по поверхности и размножаются в соответствии с их скоростью роста. Когда способность местности поддерживать виды превышена, некоторые агенты умирают, чтобы поддерживать относительно постоянную популяцию на всей территории. Модель позволяет исследовать, как разница в уровне рождаемости влияет на соотношение агентов красных и синих цветов. Интерфейс модели на сайте NetLogo Web представлен на рисунке 24.

Модель представляет достаточно простой и известный феномен, когда на одной территории существуют два близких и конкурирующих между собой вида, различающиеся только числом потомков, которые они оставляют. Возможная гипотеза исследования

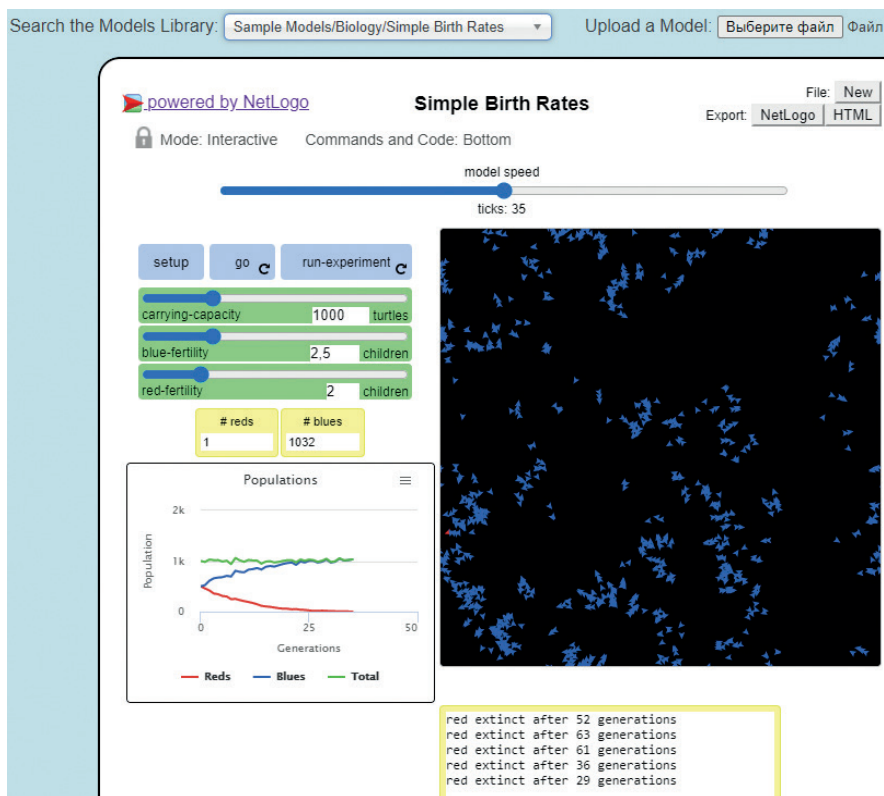


Рис. 24. Модель на сайте NetLogo Web

состоит в том, что менее плодовитый вид будет неизбежно вытеснен более плодовитым. Плодовитость красных будет оставаться неизменной (2,0), а плодовитость синих будет варьировать от 2,1 до 5,0. Чтобы проанализировать, как плодовитость синих влияет на время до исчезновения красных черепах, мы воспользуемся имитационным экспериментом, который изменяет фертильность синих от 2,1 до 5,0 с шагом 0,1, создавая в общей сложности 30 сценариев. Далее мы запустим по десять повторов каждого сценария. Каждый прогон модели будет продолжаться до тех пор, пока не исчезнут красные черепахи, а затем будет выведен номер тика, при котором происходит вымирание. Эти результаты необходимо записать в файл, который мы затем можем импортировать

в электронную таблицу или статистическое программное обеспечение. После этого нужно будет вычислить среднее время и стандартное отклонение времени до вымирания черепах красного цвета и изобразить, как они меняются в зависимости от фертильности черепах синего цвета.

По условиям внутри модели общая численность красных и синих черепах, которую может поддерживать среда обитания, остается постоянной. Мы устанавливаем общую начальную численность всех агентов равной 1000. Плодовитость красных `red-fertility` будет неизменной величиной равной двум. Плодовитость синих `blue-fertility` будет изменяться от 2,1 до 5 с интервалом в 0,1. Таким образом мы планируем провести 30 серий экспериментов и для каждой серии проведем десять повторов. В качестве конечного условия будем использовать исчезновение красных агентов: `red-count = 0`. Значение `red-count` в модели вычисляется при построении графика, отражающего численность черепах и равно `count turtles with [color = red]`.

Мы указываем формат сохранения данных — `Table output` — и получаем файл `CSV` — `BirthRate01-table.csv`, с которым в среде `R` можем проводить действия, аналогичные тем, которые мы проводили с данными о лесных пожарах.

```
bf <- read_csv("BirthRate01-table.csv", skip = 6)
```

Выберем столбцы, необходимые для построения графика

```
bf <- bf %>%
```

```
  rename (blue_fertility = 'blue-fertility') %>%
```

```
  select (blue_fertility, ticks)
```

Представим зависимость длительности существования красных черепах от плодовитости синих черепах на графике `Box Plot`.

```
boxplot(ticks ~ blue_fertility, data=bf, main="Вытеснение вида",  
        xlab="Плодовитость синих", ylab="Число ходов до гибели красных")
```

Результаты представления данных в среде `R` показаны на рисунке 25.

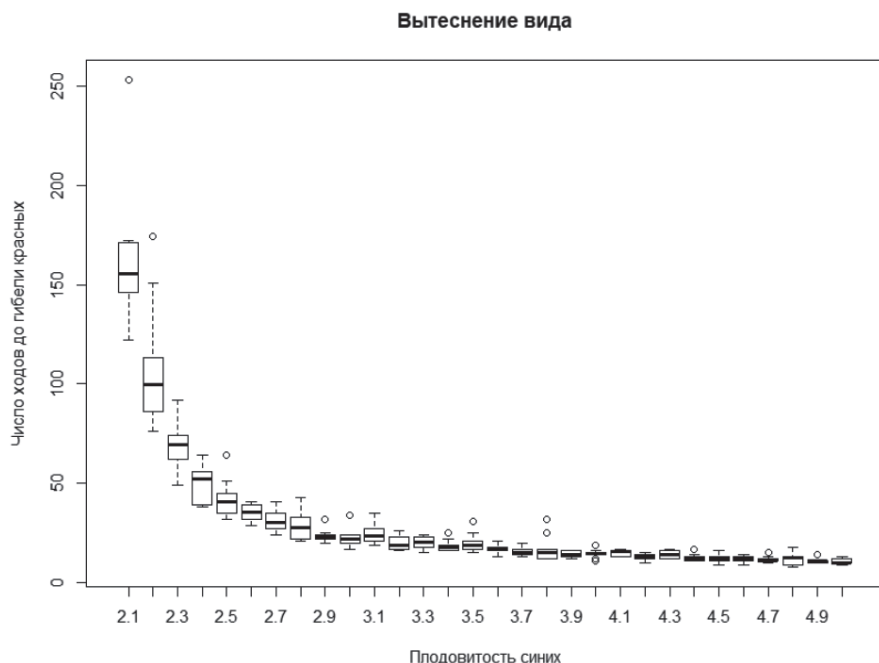


Рис. 25. Box Plot

Как отмечают авторы практического введения в агентное моделирование (Railsback, Grimm, 2019), это пример эксперимента по анализу чувствительности, в котором мы изменяем один параметр в широком диапазоне и смотрим, как модель на него реагирует. Этот экспериментальный план очень полезен для понимания того, как модель и система, которую она представляет, реагируют на изменения только одного фактора.

Мы можем сформировать правила для проведения эксперимента, изменяя параметры, управляющие поведением агентов (птиц или рыб) при образовании стаи. На рисунке 26 представлен интерфейс модели, когда мы формируем указания к инструменту, задавая правила, которым нужно следовать при проведении эксперимента.

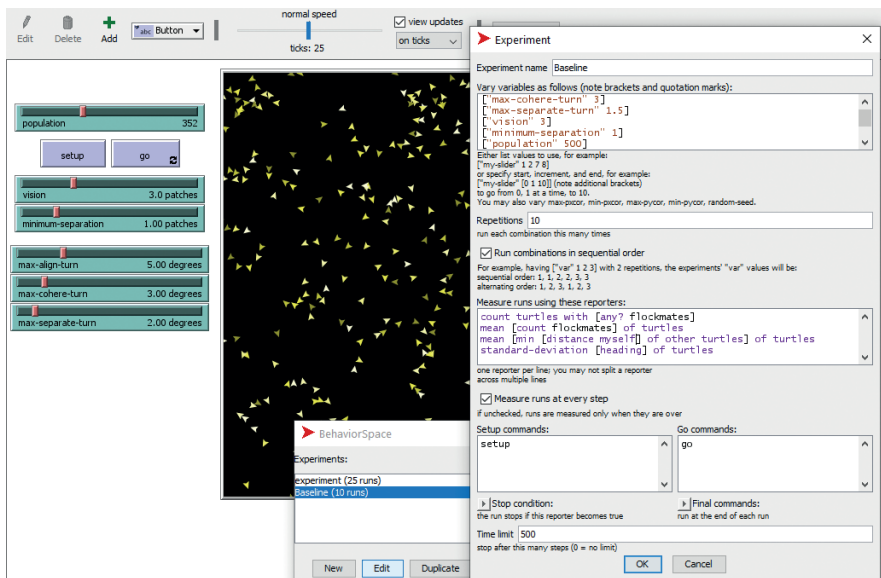


Рис. 26. Организация эксперимента с формированием стай при помощи BehaviorSpace

В первой версии каждая рыба-черепаха будет выбирать себе в товарищи по стае всех рыб, которые находятся от нее на расстоянии радиуса видения.

```
to find-flockmates
  set flockmates other turtles in-radius vision
end
```

Мы собираем данные первого эксперимента при помощи BehaviorSpace. Определяем значения независимых переменных. В данном случае они будут неизменны, поскольку свойства стаи будут меняться со временем в результате реакции агентов на поведение других агентов.

```
["max-cohere-turn" 3]
["max-separate-turn" 1.5]
["vision" 3]
["minimum-separation" 1]
["population" 500]
```

Значения, которые мы будем собирать на каждом шаге программы:

count turtles with [any? flockmates] — число птиц, у которых есть товарищи по стае
mean [count flockmates] of turtles — среднее число товарищей у каждой птицы
mean [min [distance myself] of other turtles] of turtles — среднее расстояние до других птиц
standard-deviation [heading] of turtles — стандартное отклонение в направлении от других птиц.

По завершении эксперимента мы получаем уже привычную таблицу в формате CSV и используем данные в среде R, чтобы проследить как меняются значения по мере формирования стай с предустановленными значениями независимых переменных.

А теперь мы внесем изменения в модель и поменяем правила, по которым черепахи-птицы отслеживают товарищей по стае, так что в число товарищей будет попадать только та рыба, которая ближе всех к ней расположена. Мы добавили в программу только одну строчку — set flockmates flockmates with-min [distance myself]:

```
to find-flockmates
  set flockmates other turtles in-radius vision
  set flockmates flockmates with-min [distance myself]
end
```

Никаких других изменений мы не вносим. Результаты экспериментов собираются точно так же, как они собирались для предыдущей модели при помощи BehaviorSpace. В данном случае нам важно показать, как можно объединять и сравнивать результаты двух экспериментов.

```
v1 <- read_csv("Flocking_experiment_1_Baseline-table.csv", skip = 6)
v2 <- read_csv("Flocking_experiment_2_Baseline-table.csv", skip = 6)
v1 <- v1 %>%
  rename(Tick = '[step]', with_mates = 'count turtles with [any? flockmates]', mean_mates = 'mean [count
```

```

flockmates] of turtles', min_dist = 'mean [min
[distance myself] of other turtles] of turtles',
heading_dev = 'standard-deviation [heading] of turtles') %>%
  select(Tick, with_mates, mean_mates, min_dist, heading_dev)

v2 <- v2 %>%
  rename(Tick = '[step]', with_mates = 'count turtles
with [any? flockmates]', mean_mates = 'mean [count
flockmates] of turtles', min_dist = 'mean [min
[distance myself] of other turtles] of turtles',
heading_dev = 'standard-deviation [heading] of turtles') %>%
  select(Tick, with_mates, mean_mates, min_dist ,
heading_dev)

p <- ggplot(v1, aes(x = Tick, y = heading_dev) ) +
  geom_smooth(fill="blue", colour="darkblue", size=1) +
  scale_x_continuous(name = "Число ходов (ticks)") +
  scale_y_continuous(name = "Отклонение от общей направ-
ленности (heading)")
p <- p + geom_smooth(data=v2, fill="red", colour=
"darkred", size=1)
print(p)

```

Динамика того, как меняется число агентов, у которых есть товарищи по стае, представлена на рисунке 27. Синим цветом на графике показаны результаты эксперимента, когда товарищами по стае считаются все агенты, которые находятся в радиусе видимости. Красным цветом показаны результаты, когда товарищем по стае считается только ближайший агент.

Динамика среднего числа товарищей у каждой черепашки представлена на рисунке 28. Синим цветом на графике показаны результаты эксперимента, когда товарищами по стае считаются все агенты, которые находятся в радиусе видимости. Красным цветом показаны результаты, когда товарищем по стае считается ближайший агент. Здесь ожидаемо товарищей по стае не может быть больше одного агента.

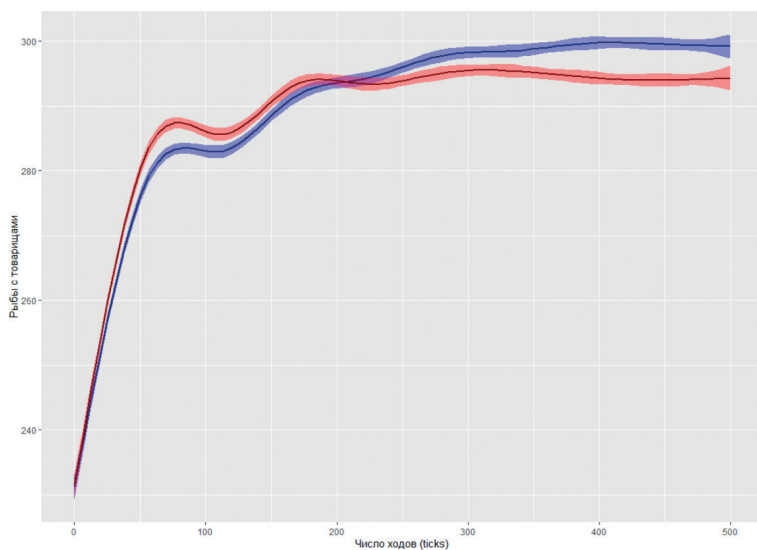


Рис. 27. Сравнение экспериментов по изменению числа flockmates

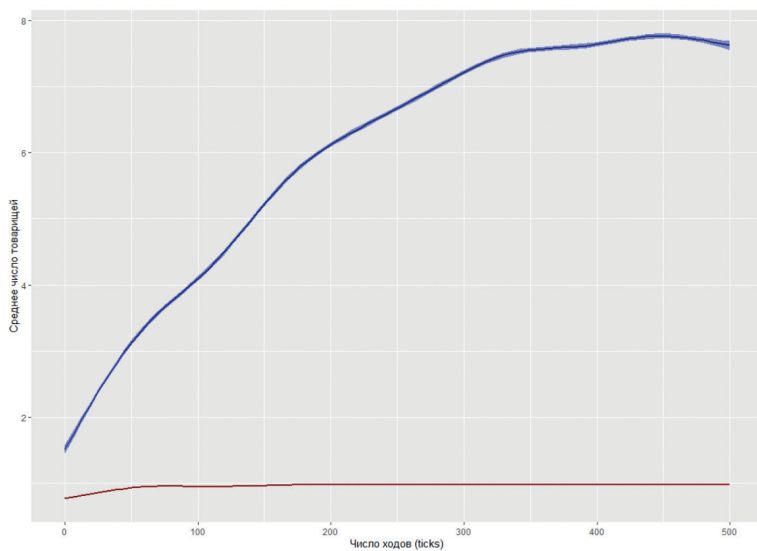


Рис. 28. Сравнение экспериментов по среднему числу товарищей

Динамика среднего расстояния до других агентов представлена на рисунке 29. Синим цветом на графике показаны результаты эксперимента, когда товарищами по стае считаются все агенты, которые находятся в радиусе видимости. Красным цветом показаны результаты, когда товарищем по стае считается ближайший агент.

Мы видим, что среднее расстояние между агентами уменьшается и они становятся ближе друг к другу независимо от того, какая стратегия выбора товарищей по стае используется в эксперименте.

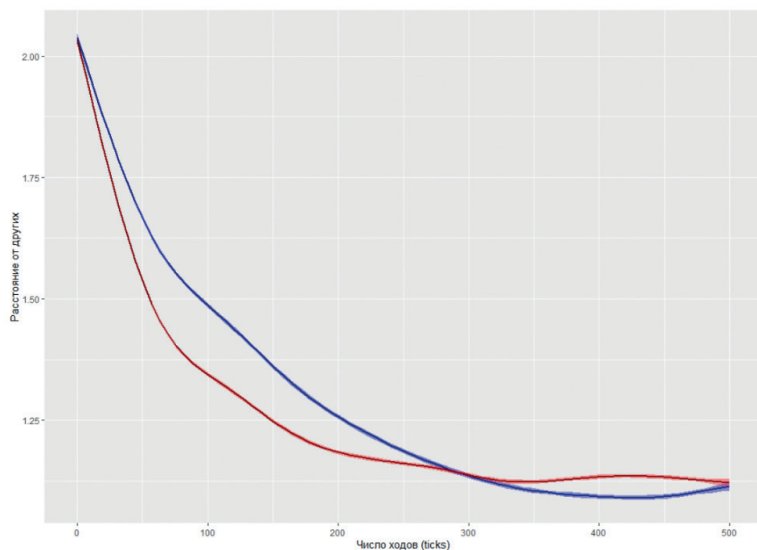


Рис. 29. Сравнение экспериментов по среднему расстоянию до других рыб

Динамика стандартного отклонения в направлении от других агентов представлена на рисунке 30. Синим цветом на графике показаны результаты эксперимента, когда товарищами по стае считаются все агенты, которые находятся в радиусе видимости. Красным цветом показаны результаты, когда товарищем по стае считается ближайший агент.

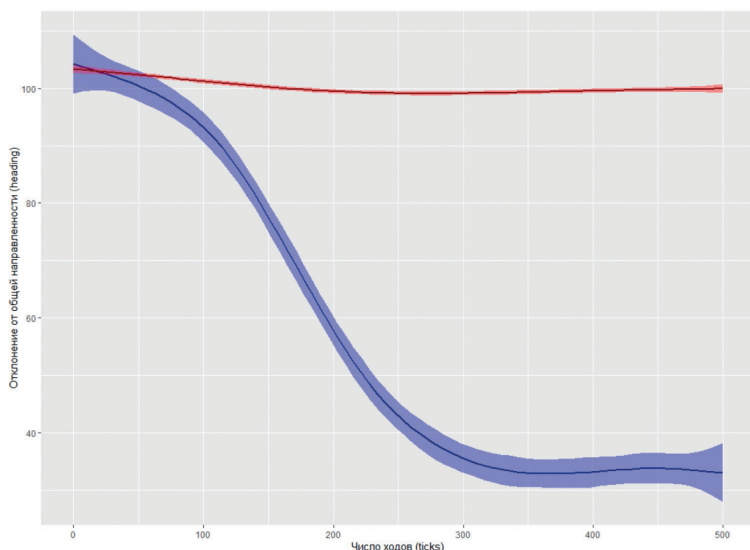


Рис. 30. Сравнение экспериментов по отклонению от общего направления

Мы уделили внимание рассмотрению возможностей экспериментов с моделью образования стай, поскольку эта модель часто используется в качестве отправной точки для изучения процессов самообразования команд и экспертных групп. Так, Томас Вилл утверждает, что различные элементы модели стаеобразования в среде NetLogo — двумерное пространство, правила взаимодействия, коллективные результаты — имеют аналоги в контексте человеческой организации (Will 2016; Will 2016a). Агенты в стайных моделях перемещаются через участки двумерного пространства. В любой конкретный момент времени агент занимает определенное положение и имеет заданное направление. В контексте человеческой организации нас интересует не то, как люди относятся друг к другу физически, а скорее то, как они относятся друг к другу в ходе коллективной работы над проблемой. Вилл рассматривает каждый фрагмент двумерного пространства как интерпретацию любого вызова, стоящего перед рассматриваемым человеческим коллективом. Направленность человека отражает последовательность,

предсказуемость и направление развития его индивидуального способа восприятия реальности. Внешняя среда, в которой действует коллектив и с которой он взаимодействует, представлена в стайной модели скорее косвенно, чем напрямую. Внешние стимулы опосредуются индивидуальным восприятием и реакцией. Другими словами, среда выражается через ее влияние на индивидуальные познания и перспективы. Как общие правила поведения, нормы группового взаимодействия используются в контексте социума, так правила взаимодействия работают в модели образования стаи. Теория лидерства в стае исследует, как проявляются различные способности к коллективному обучению, когда взаимодействующие люди работают над решением проблем. Она моделирует коллективное обучение, понимаемое как изменение способов восприятия реальности и реагирования на нее, как движение в двумерном пространстве. Прецеденты моделей обучения как движения есть и в управленческой, и в психологической литературе. Модель стайного лидерства исследует динамику коллективного поиска интерактивного обучения, которая и ранее исследовалась как движение агента в двумерном пространстве (Gabora, Saberi, 2011). При этом авторы использовали в качестве отправной точки идею ремиксов, когда одна идея вызывает другую, исправленную версию, которая вызывает еще одну, и так далее рекурсивно. Их модель вращается вокруг трех основных элементов коллективного когнитивного развития: индивидуального изобретения, оценки сверстниками и имитации сверстников. Эти три элемента сходны со склонностью агентов в стайных моделях отделяться от других и согласовывать свои действия с действиями других агентов.

Одна из первых работ, на базе которой строились последующие модели и симуляции, это статья Гуимера и соавторов (Guimerà et al. 2005), в которой они проанализировали варианты формирования команд, обратив внимание на то, что людям свойственно сотрудничать с теми, с кем у них уже был опыт сотрудничества. Те, кто уже снимался в фильме или писал статью, предпочитают в последующих проектах сотрудничать с проверенными соавторами. В дальнейшем в предложенную модель вносились изменения

и дополнения, но всегда сохранялась общая направленность на поиск и объяснение того, почему люди объединяются в творческие команды и какие факторы влияют на выбор тех, с кем они создают общий продукт.

Многие общие черты, присущие сетям творческих предприятий, можно уловить с помощью модели Team Assembly с двумя простыми параметрами: долей новичков, участвующих в команде, и склонностью бывших сотрудников снова работать друг с другом.

При каждом цикле (тике) собирается новая команда. Члены команды — либо неопытные новички — люди, которые ранее не участвовали ни в каких командах, либо уже признанные «действующие лица» — опытные люди, которые ранее уже участвовали в команде. Каждый член выбирается последовательно.

Ползунок P показывает вероятность того, что новым членом команды станет опытный участник.

```
repeat team-size
[
  ifelse random-float 100.0 >= p ;;with a probability P,
  make a newcomer
  [
    make-newcomer
    set new-team-member newcomer
  ]
]
```

Если новый участник не является новичком, то с вероятностью, заданной ползунком Q, новый опытный участник будет выбран случайным образом из пула предыдущих сотрудников того опытного участника, который уже входит в команду. В противном случае новый член будет просто выбран случайным образом из списка всех опытных.

```
ifelse random-float 100.0 < q and any? (turtles
with [in-team? and (any? link-neighbors with [not
in-team?])])
[set new-team-member one-of turtles with [not in-team?
and (any? link-neighbors with [in-team?])]]
[set new-team-member one-of turtles with [not in-team?]]
```

При создании команды все участники связаны друг с другом.

Если агент не участвует в новой команде в течение длительного периода времени, агент и его ссылки удаляются из сети.

```
ask turtles
[ if downtime > max-downtime [die] ]
```

Агенты в недавно собранной команде отмечены синим цветом, если они новички, и желтым, если опытные. Маленькие серые кружки обозначают тех, кто в настоящее время не сотрудничает. Ссылки указывают на опыт участников во время их самого последнего сотрудничества. Например, синие ссылки между агентами указывают на то, что два агента сотрудничали как новички. Зеленые и желтые ссылки соответствуют одноразовому сотрудничеству новичка и опытного, а также опытного и опытного соответственно. Наконец, красные ссылки указывают на то, что агенты сотрудничали друг с другом несколько раз.

С моделью можно экспериментировать, она доступна в библиотеке NetLogo и на сайте NetLogo Web.

Интерфейс модели образования команд на сайте NetLogo представлен на рисунке 31.

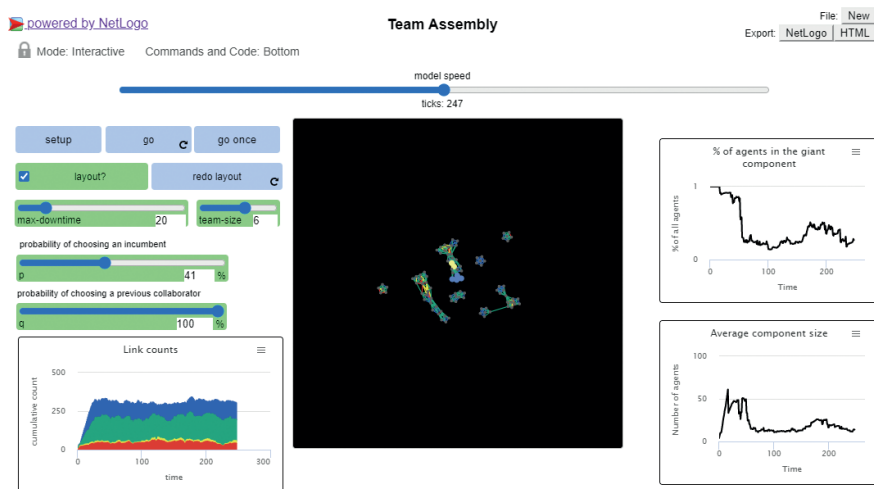


Рис. 31. Интерфейс модели формирования команд

Модель NetLogo позволяет увидеть на экране различные возникающие топологии связей между участниками. Новички и новые команды всегда появляются в центре экрана. Команды или кластеры команд, не связанные с новыми участниками, постепенно смещаются к краю экрана. Опытные участники, выбранные случайным образом, могут располагаться в любой части экрана. Таким образом, сотрудничество между новичками происходит в центре экрана, а разъединенные кластеры отталкиваются от центра экрана.

Структура сети совместной деятельности в модели может резко измениться со временем. Изначально создаются только команды новичков, поскольку поле сотрудничества существует недостаточно долго, чтобы участники могли покинуть поле совместной деятельности. Однако по прошествии некоторого периода времени (MAX-DOWNTIME) неактивные агенты начинают удаляться из игры и количество агентов становится относительно стабильным — возникающие эффекты Р и Q оказываются более очевидными на этой стадии равновесия. Кроме того, по окончании стадии роста часто отмечается падение связанности сети.

Модель отражает две основные особенности сетей сотрудничества, которые могут влиять на распространение знаний внутри образовательного сообщества, ускоряя или сдерживая распространение инноваций внутри организации простым изменением значения Р и Q.

Во-первых, это распределение различных типов связей между участниками, которое можно увидеть на графике LINK COUNTS.

```
let total 0
set-current-plot-pen "previous collaborators"
plot-pen-up plotxy ticks total
set total total + count links with [color = red]
plot-pen-down plotxy ticks total
set-current-plot-pen "incumbent-incumbent"
plot-pen-up plotxy ticks total
set total total + count links with [color = yellow]
plot-pen-down plotxy ticks total
set-current-plot-pen "newcomer-incumbent"
plot-pen-up plotxy ticks total
```

```
set total total + count links with [color = turquoise]
plot-pen-down plotxy ticks total
set-current-plot-pen "newcomer-newcomer"
plot-pen-up plotxy ticks total
set total total + count links with [color = blue]
plot-pen-down plotxy ticks total
```

Переизбыток связей «новичок – новичок» (синие связи) может указывать на то, что в этой области совместной деятельности экспертные группы не используют преимущества опытных участников. С другой стороны, избыток повторных взаимодействий (связи красного цвета) или связей между опытными участниками (связи желтого цвета) могут указывать на отсутствие новизны и разнообразия в публикуемых проектах или сценариях уроков.

Во-вторых, это общая связность сети совместной деятельности. Например, утверждается, что многие академические области состоят из гигантского «невидимого колледжа», в который включены слабо связанные между собой академические сообщества. Напротив, патентные сети, как правило, состоят из изолированных кластеров или цепочек изобретателей. Процент участников, относящихся к гигантскому компоненту, можно увидеть на графике % OF AGENTS IN THE GIANT COMPONENT: гигантский компонент равен размеру самой большой связанной цепочки сотрудников:

```
plotxy ticks (giant-component-size / (count turtles))
```

При описании модели совместной деятельности авторы указывают, что формируемая сеть взаимодействия чаще всего представляет собой бимодальную сеть, в которой объединение участников происходит через объекты, которые они создают: ученые и их публикации, ученики и их проекты, учителя и сценарии уроков. Связь между учеными и публикациями означает авторство ученого по отношению к публикации. Связи между одной публикацией и несколькими учеными указывают на их соавторство по отношению к этой публикации.

Модель формирования команд неоднократно видоизменялась и подстраивалась под особенности конкретных ситуаций. Например, в работе 2019 года авторы использовали ее для изучения

исследовательской активности внутри национального института (García-Peña et al. 2019).

Многоагентные модели изучения процессов создания и использования цифрового контента, как правило, опираются на предшествующие модели, в которых агенты взаимодействовали между собой опосредованно, создавая или изменяя некий субстрат. Многочисленные примеры таких моделей есть в библиотеке моделей, распространяемых вместе с языком NetLogo, на сайте NetLogo Commons.

Наибольший интерес для целей данного исследования представляют те из этих моделей, в которых имитируется взаимодействие множества агентов, образующих стаи, команды и коллективы. Среди выбранных моделей, которые можно использовать в качестве примеров и источников исходных кодов, отметим следующие:

- <http://ccl.northwestern.edu/netlogo/models/Flocking> — модель формирования стаи (птиц, рыб и т. п.), двигающейся в одном направлении благодаря тому, что каждый агент подстраивается под поведение окружающих его агентов;

- <http://ccl.northwestern.edu/netlogo/models/Cooperation> — модель возникновения кооперации, когда агенты используют субстрат, только когда он подрос;

- <http://ccl.northwestern.edu/netlogo/models/Termites> — модель термитов — особенно важна для нашего исследования, поскольку взаимодействие агентов в этой модели опосредовано объектами-палочками, которые они собирают; <http://ccl.northwestern.edu/netlogo/models/TeamAssembly> — формирование команд — интерес представляет использование в свойствах агентов такого параметра, как возраст. По достижении определенного возраста агент перестает участвовать в формировании команд;

- <http://ccl.northwestern.edu/netlogo/models/Sugarscape3WealthDistribution> — вариант сахарной модели, где действия агентов связаны с потреблением и накоплением сахара как богатства;

- <http://ccl.northwestern.edu/netlogo/models/PreferentialAttachment> — предпочтительное присоединение, — когда новые узлы добавляются преимущественно к тем, у кого уже больше связей.

ГЛАВА 3

ПРИМЕРЫ АНАЛИЗА СОВМЕСТНОЙ ДЕЯТЕЛЬНОСТИ

В данной главе мы анализируем цифровые следы, которые оставляли участники совместной деятельности в электронных библиотеках, в различных категориях вики-проектов, в студиях и коллекциях различных сообществ обучения программированию, внутри школьных доменов при совместном создании документов, в цифровых образовательных репозиториях.

3.1. Результаты библиометрического картирования поля исследования

На поле исследования мы выделяем два взаимосвязанных пространства: 1) тематику совместной сетевой деятельности и 2) тематику формирования вычислительного мышления. Мы использовали методы библиографического анализа баз данных научного цитирования Web of Science, Scopus и ACM Digital Library. В качестве средства библиографического анализа был выбран VOSviewer — средство, которое динамично развивается с 2009 года (Eck, Waltman 2009) и используется для выявления и визуального представления сетевых отношений как между авторами, так и между ключевыми словами публикаций (Williams 2020). Следует отметить, что этот инструмент — один из многочисленных инструментов, которые были разработаны специально для макроскопических исследований в области командной науки. Например, развернутую аналитику по выбранным публикациям можно получить с помощью пакета Bibliometrix в среде R. Подробное описание метода библиометрического картирования представлено в статье об исследовании научных школ средствами библиометрического картирования (Дудко, Патаракин 2023).

Запрос к библиографической базе данных Web of Science был сформулирован так, чтобы получить все документы типа «статья», связанные с темой формирования команд и темой командной науки, которые были написаны на английском языке в последние 20 лет (2001–2021 гг.).

(TS=(("Team Science" OR "Team Formation" OR "Team Assembly" OR "Invisible College") AND (computer OR internet OR digital OR virtual))) AND LANGUAGE: (English) AND DOCUMENT TYPES: (Article)
Timespan: 2001–2021. Indexes: SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH, BKCI-S, BKCI-SSH, ESCI

В результате поиска мы получили 92 работы из Web of Science Core Collection. Использование VOSviewer позволило выделить кластеры понятий, которыми пользуются авторы, изучающие феномен командной научной деятельности (рис. 32).

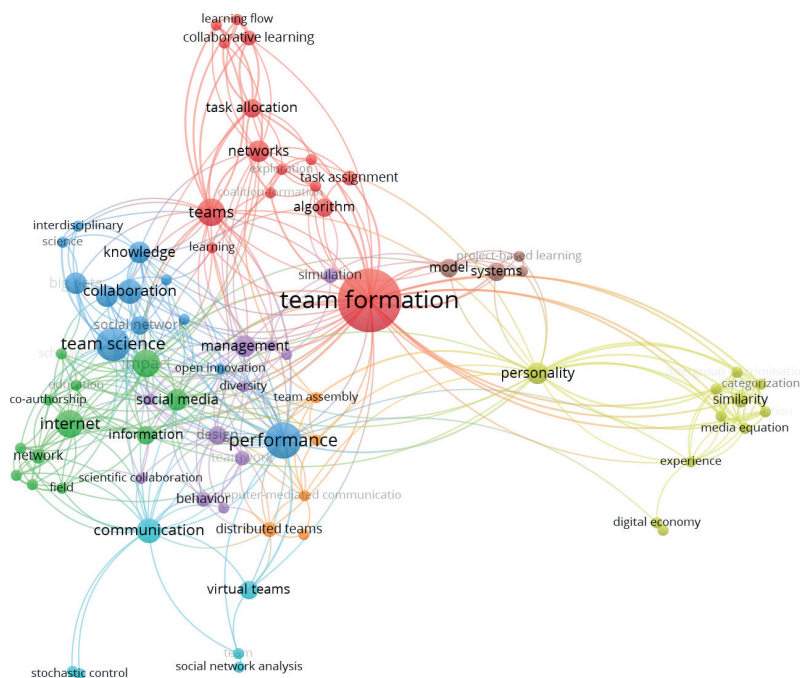


Рис. 32. Кластеры понятий, связанных с группами командной науки (WoS)

Наиболее крупный красный кластер содержит 15 ключевых слов: команды, формирование команд, многоагентные системы, сети, обучение, постановка задачи, оценка задачи, формирование коалиций, стили обучения, процесс обучения.

Второй, зеленый, кластер содержит 12 ключевых слов: соавторство, Интернет, сетевой анализ, «невидимый колледж», сеть, паутина, социальные медиа.

Третий, синий, кластер содержит 12 слов: большие данные, взаимодействия, инновации, знания, выполнение, наука, командная наука.

Кроме определения кластеров ключевых слов, представляющих область науки о командах и экспертных группах, мы выделили и сами экспертные группы ученых, которые объединены в «невидимый колледж» через публикации, размещенные в базе данных Web of Science. Общее поле, на котором представлены группы участников, можно увидеть на рисунке 33. Мы видим небольшие группы участников, которые никак не связаны друг с другом.

Запрос к библиографической базе данных Scopus был сформулирован так, чтобы получить все документы (статьи, книги, главы в книгах), связанные с темой формирования команд и темой командной науки, которые были написаны на английском языке в последние 20 лет (2001–2021 гг.) в области социальных или компьютерных наук:

```
ALL ( "Team Science" OR "Team Formation" OR "Team Assembly" OR "Invisible College" ) AND PUBYEAR > 2001 AND ( LIMIT-TO ( PUBSTAGE , "final" ) ) AND ( LIMIT-TO ( OA , "all" ) ) AND ( LIMIT-TO ( DOCTYPE , "ar" ) OR LIMIT-TO ( DOCTYPE , "ch" ) OR LIMIT-TO ( DOCTYPE , "bk" ) ) AND ( LIMIT-TO ( SUBJAREA , "SOCI" ) ) AND ( LIMIT-TO ( LANGUAGE , "English" ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) )
```

В результате поиска по заданным критериям мы получили 205 документов. Использование VOSviewer позволило выделить кластеры понятий, которыми пользуются авторы, изучающие феномен командной научной деятельности (см. рис. 34).

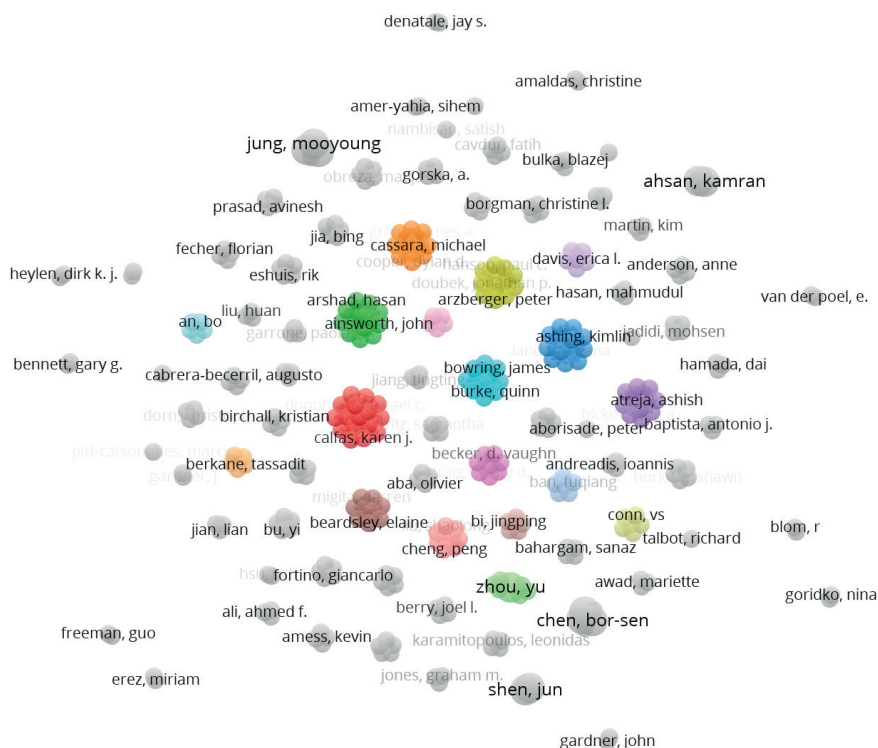


Рис. 33. Кластеры участников с общей экспертной позицией (Web of Science)

Кроме определения кластеров ключевых слов, представляющих область науки о командах и экспертных группах, мы выделили и сами экспертные группы ученых, которые объединены в «невидимый колледж» через публикации, размещенные в базе данных Scopus. Общее поле, на котором представлены группы участников, можно увидеть на рисунке 35.

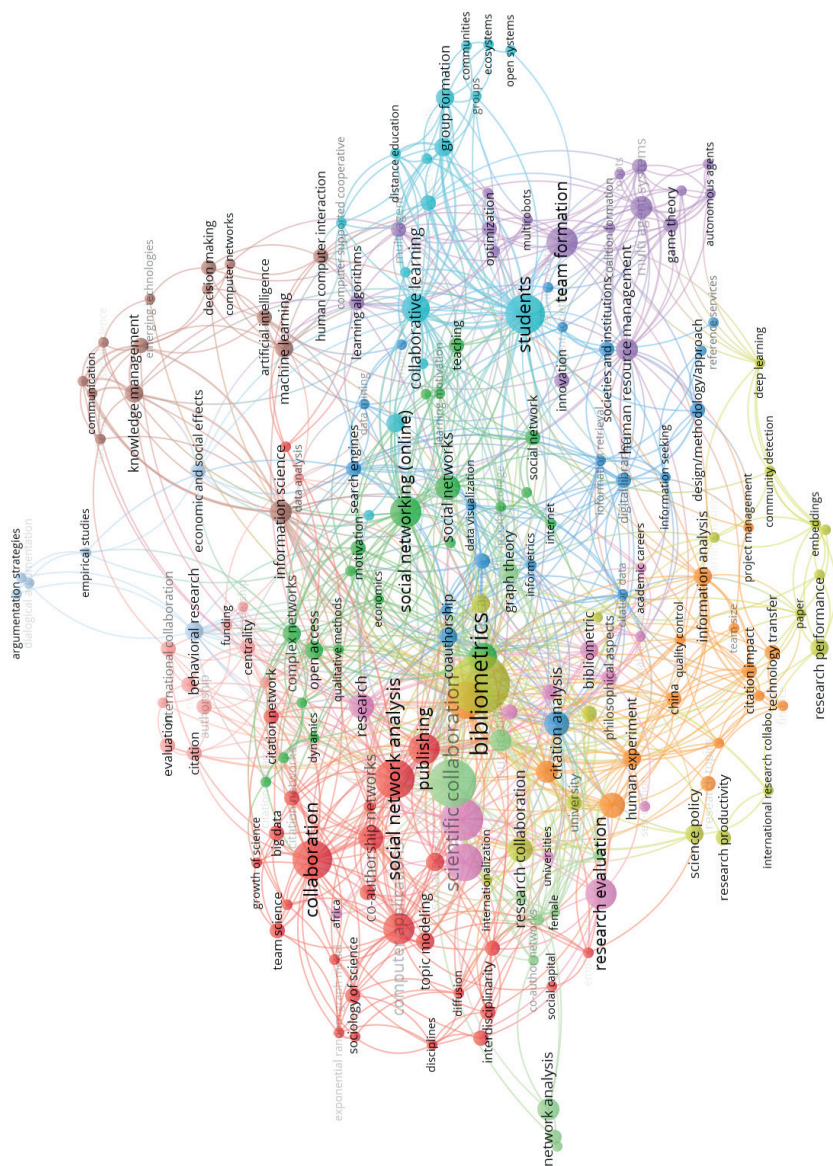


Рис. 34. Кластеры понятий, связанных с группами командной науки (Scopus)

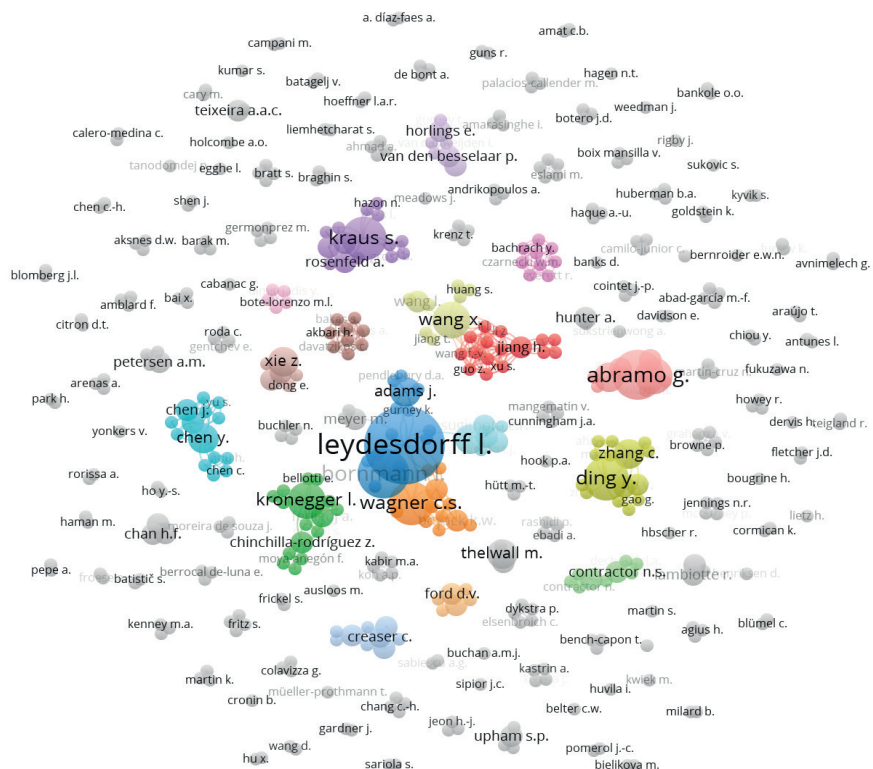


Рис. 35. Кластеры участников с общей экспертной позицией (Scopus)

В дальнейшем мы провели краткий анализ исследований по наиболее крупным кластерам Scopus и Web of Science, связанных с выявлением команд и экспертных групп. В рамках последующей работы мы обновили данные о поле понятий и об авторах, которые пишут о проблемах совместной работы со знаниями в образовании. Для такого обновления мы ограничили сферу поиска только материалами цифровой библиотеки Ассоциации вычислительной техники как самой специализированной библиотеки крупнейшей международной организации в области информационных технологий — ACM (dl.acm.org). Мы ограничили временной период написания статей последними десятью годами, поскольку более ранние статьи были изучены нами в предыдущих работах.

Поисковый запрос выглядит следующим образом:

```
[All: "collaborative knowledge"] AND [Publication  
Date: (2012 To 2022)]
```

Мы проверили разницу между этим запросом и запросом collaborative knowledge and education и обнаружили, что разница в количестве ответов была несущественной. Поскольку мы боялись потерять статьи, в которых была бы представлена перспективная форма работы со знаниями, то мы отказались от использования этого фильтра. Однако на исходный запрос мы наложили фильтр, ограничивающий формат статей только исследовательскими статьями, поскольку мы хотели получить подробное описание кейсов, в которых описывалась бы практика работы со знаниями в образовании. В ответ на такой запрос было получено 476 записей, которые составили коллекцию в системе Zotero и в формате RIS были загружены в систему VOSviewer для выявления и визуального представления сетевых отношений как между авторами, так и между ключевыми словами публикаций. Сеть ключевых понятий, представляющих поля публикаций, посвященных совместной деятельности, представлена на рисунке 36.

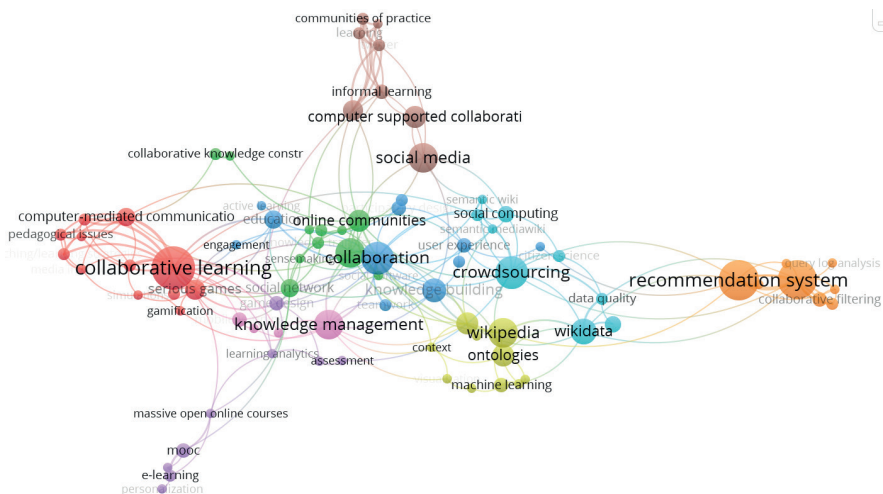


Рис. 36. Сеть понятий поля совместной работы со знаниями

Наиболее многочисленным является красный кластер, в котором представлены совместное обучение, геймификация и серьезные игры. Зеленый кластер объединяет сетевые сообщества, социальные сети, совместную работу и передачу знаний. Красный и зеленый кластеры связаны между собой совместным конструированием знаний. Близко к зеленому и красному кластерам расположен розовый кластер управления знаниями, через который эти два кластера связаны с фиолетовым кластером учебной аналитики, оценивания и массовыми онлайн-курсами. Рядом с зеленым кластером расположен синий кластер взаимодействия и построения знаний. Этот синий кластер связан со светло-зеленым кластером Википедии, онтологий, контекста и машинного обучения. Рядом с этим кластером расположен голубой кластер данных, вики-данных и семантических вики. С кластером данных связан золотой по цвету кластер рекомендательных систем и коллаборативной фильтрации. Сверху от зеленого кластера сетевых сообществ расположен коричневый кластер сообществ практики и взаимодействия при помощи компьютеров. Все кластеры связаны между собой. Поле понятий, сформированное на основе анализа публикаций из цифровой библиотеки ACM, последовательно обработанное в менеджерах Zotero и VOSviewer доступно читателям в форме интерактивного апплета по адресу: <https://app.vosviewer.com/?json=https://drive.google.com/uc?id=1ApioSuBY2SYSF9ZfEo6IIoG3QuvZmIQu>

В качестве основного источника данных для библиометрического анализа взаимоотношений авторов, связанных с тематикой вычислительного мышления, мы использовали материалы цифровой библиотеки Ассоциации вычислительной техники (dl.acm.org). Выбор этого источника связан с тем, что цифровая библиотека в течение почти семидесяти лет целенаправленно собирает работы, связанные с использованием компьютеров в различных сферах человеческой деятельности и здесь ожидаемо можно проследить историю обсуждения понятия «вычислительное мышление», различные подходы к вычислительному мышлению, связь вычислительного мышления с языками обучения программированию и видеоиграми.

Поисковый запрос к цифровой библиотеке ACM computational thinking с ограничением временного периода 2002–2022 гг. выглядит следующим образом:

```
[All: "computational thinking"] AND [Publication Date:  
(2002 To 2022)]
```

Этот запрос позволил получить 1680 записей в формате BIB о публикациях по теме вычислительного мышления. Обработка — слияние данных, проверка записей и удаление дубликатов — осуществлялась в библиографическом менеджере Zotero (Cai et al., 2021; Fernandez, 2011), поскольку он поддерживает различные формы работы с исходными текстами публикаций, обеспечивает экспорт из различных библиотек и импорт в различные системы анализа и представления библиографических данных. После удаления дубликатов число записей составило 1645. Для выявления и визуального представления сетевых отношений как между авторами, так и между ключевыми словами публикаций (Williams 2020) мы использовали пакет VOSviewer (Eck, Waltman 2009), предварительно экспортировав их из Zotero в формат RIS.

Для анализа ключевых понятий мы отобрали понятия, которые встречаются не менее чем в пяти статьях, и собрали тезаурус, который позволил объединить идентичные по смыслу ключевые слова. Кроме того, были удалены пять наиболее популярных и очевидных ключевых слов: вычислительное мышление, k12, изучение информатики, программирование, информатика, — чтобы разбить сеть на хорошо различимые кластеры.

Поле ключевых понятий, включающее восемь разноцветных кластеров, представлено на рисунке 37.

В красный кластер входят язык программирования Scratch; составляющие вычислительного мышления: «решение проблем», «абстракция» и алгоритмизация; блочное программирование; программирование без компьютера; обучение и введение в программирование; оценивание, учебная аналитика и масштабируемый дизайн игр.

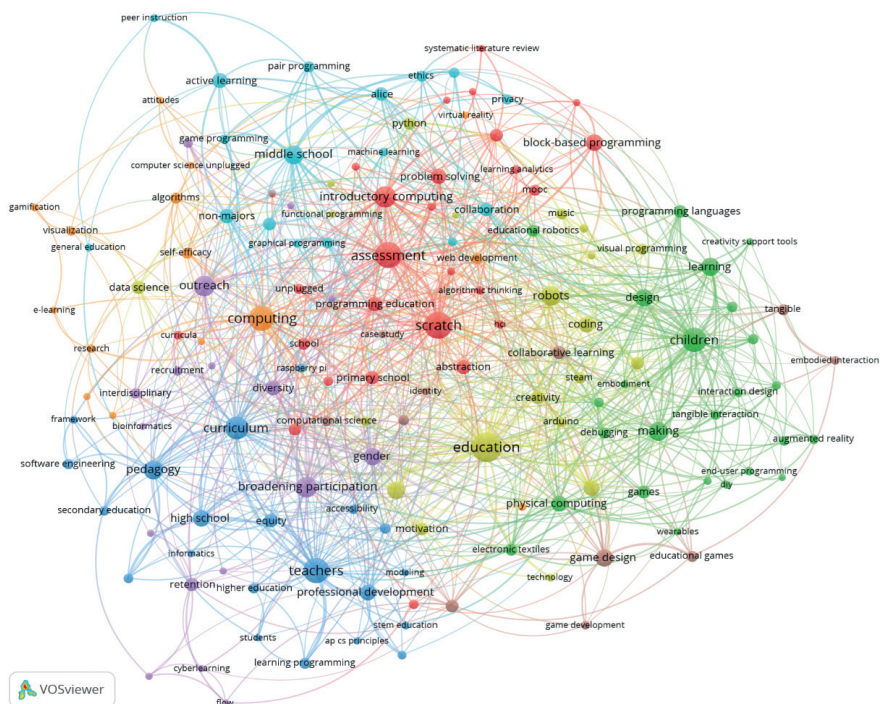


Рис. 37. Ключевые слова, представляющие поле
вычислительного мышления

В темно-зеленом кластере — дети, дизайн, учеба, мейкинг, осязаемое программирование, дополненная реальность, носимые вычисления, цифровое производство.

Синий кластер — учителя, ученики, программы учебных курсов, профессиональное развитие, различные уровни образования, принципы обучения компьютерным дисциплинам.

В темно-желтом кластере собраны такие понятия, как образование, информатика, роботы, визуальное программирование, конструкторизм, кодирование, STEAM, наука о данных, искусство, ардуино, язык Python, язык Logo, мотивация.

В фиолетовом кластере собраны такие понятия, как расширение охвата, гендер, разнообразие, удержание, междисциплинарность, привлечение, поток, app inventor.

Бирюзовый кластер — Alice, активное изучение, дополнительное образование, сотрудничество, большие данные, начальная школа, средняя школа, парное программирование, этика, игровое обучение, конфиденциальность, программирование игр, искусственный интеллект, машинное обучение.

Оранжевый кластер — компьютерные вычисления, вычислительная грамотность, веб-разработка, самоэффективность, алгоритмы, визуализация, исследование, конкурс Bebras.

Коричневый кластер — игровой дизайн, оценивание, материальное программирование, математика, образовательные игры, разработка игр. На общей карте вычислительного мышления игры представлены только внутри данного кластера.

Мы рассмотрели также сеть авторов, которые пишут статьи по тематике вычислительного мышления. Сеть авторов, опубликовавших не менее двух статей по этой тематике представлена на рисунке 38. Формат печатной монографии не позволяет читателям самостоятельно исследовать группы авторов, но это можно сделать при помощи интерактивного апплета, который доступен по адресу: https://app.vosviewer.com/?json=https://drive.google.com/uc?id=1iRge_3lB6v0yTtFPFR0EE6lfr-EKhROO

Следует обратить внимание, что группы авторов связаны между собой и даже при том, что они аффилированы с различными институтами, в целом мы наблюдаем феномен образования «невидимого колледжа» исследователей вычислительного мышления (Sedita et al. 2020).

А теперь мы рассмотрим, почему образовались именно такие кластеры понятий и как с этими кластерами связаны группы авторов. Мы начнем рассмотрение с сети понятий, представляющих карту вычислительного мышления, и постараемся проинтерпретировать объединение понятий именно в такие кластеры.

Наиболее многочисленный красный кластер выстраивается вокруг языка блочного программирования Scratch, который является наиболее распространенным языком для введения в программирование на уровне начального образования и объединяет более 75 млн пользователей во всем мире. Присутствие в красном кластере

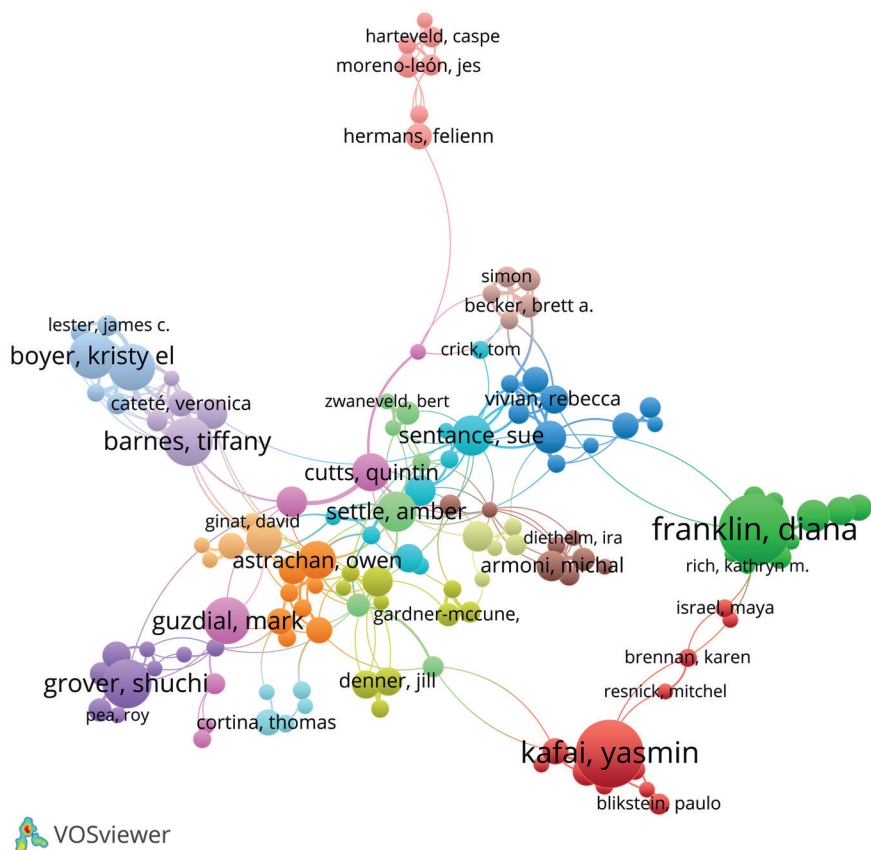


Рис. 38. Сеть авторов, исследующих понятие
вычислительного мышления

понятий, связанных с учебной аналитикой и оцениванием, легко объяснимо. Огромное число публикуемых проектов, объединенных между собой через видеозменения — ремиксы, — открывает возможности для учебной аналитики, данные которой зачастую доступны и самим участникам учебной деятельности. Современные исследователи вычислительного мышления зачастую стремятся разложить его на отдельные составляющие, проверять работы учеников по заранее определенным критериям. Перечень таких критериев: контроль, представление данных, абстракция, интерактивное

взаимодействие, синхронизация, параллельность действий и логика — обсуждался в работе (Patarakin et al. 2019) и выглядит удобным с точки зрения учителя, который хочет проверить умения ученика, и с точки зрения ученика, который хочет получить сертификат, что он владеет перечнем компетенций и его проект отражает высокий уровень этих компетенций. На первый взгляд, неожиданным выглядит присутствие в красном кластере масштабируемого дизайна игр, но это объясняется тем, что этот подход реализуется на базе близких Scratch языков блочного программирования AgentSheets и AgentCubes, и тем вниманием, которое разработчики этого направления уделяют процедуре оценивания.

В сети авторов, исследующих вычислительное мышление (см. рис. 38), красному кластеру рисунка 37 соответствуют три группы авторов, расположенные в верхней части рисунка. Первая, самая верхняя, группа, окрашенная в светло-зеленый цвет, ключевой фигурой которой на схеме является Александр Репеннинг (блочные языки программирования, масштабируемый игровой дизайн, теория зоны ближайшего потока) (Repenning 2013; Repenning, Basawapatna 2016). Рядом с первой группой расположена вторая группа, также окрашенная в светло-зеленый цвет. Ключевым игроком в этой группе на схеме является Ясмин Кафаи (Kafai 2016). Исходно авторы этой группы связаны с MIT Media Lab, разработкой различных версий программной среды Logo, наиболее успешным примером которой является среда Scratch. Через ключевого игрока Карен Бреннан (Brennan, Resnick 2013) эта группа связана с оранжевым кластером, в центре которого находится Диана Франклин (Franklin 2015). Необходимо отметить, что связующим игроком, объединяющим оранжевый кластер с другими расположенными ниже группами, является создатель языка NetLogo Ури Виленский (Weintrop, Wilensky 2015).

Второй по численности темно-зеленый кластер — осязаемое, носимое и деятельностное освоение детьми вычислительных технологий, языков программирования и средств творческой деятельности. Внутри темно-зеленого кластера-сообщества наиболее популярным понятием является «делание» — мейкинг, а другие термины

его дополняют и расшифровывают. Здесь трудно провести прямое соответствие между кластером понятий на рисунке 37 и кластерами авторов на рисунке 38, поскольку активно сотрудничают авторы, которые пишут и про вычислительное участие, и про использование электронных тканей — Ясмин Кафай и группа ее аспирантов. С этой группой связан и Паоло Бликштейн (Blikstein 2013; Blikstein, Krannich 2013), известный своим анализом мейкерского движения в образовании и организацией мейкерских центров FabLab внутри школ.

Очень близко со вторым, темно-зеленым, кластером расположен четвертый, темно-желтый, кластер. Многие понятия темно-желтого кластера дополняют темно-зеленый: роботы и конструкторы ардуино, программируемые на языке Python и ряде логоподобных языков; искусство, расширяющее понятие STEM (наука, технологии, инженерия и математика) до STEAM (наука, технологии, инженерия, искусство и математика). К этим мейкерским понятиям добавляется теоретический конструкт «конструкционизм», который Бликштейн и его коллеги рассматривают как основание для создания языков и визуального, и блочного программирования и для мейкерского движения.

Третий, синий, кластер связывает понятия учителей, учеников, программы учебных курсов, STEM-образование, профессиональное развитие педагогов и общие принципы обучения компьютерным дисциплинам. Для этого кластера сложно выделить отдельную группу авторов на рисунке 38, поскольку про STEM-образование, учебные планы и курсы обучения пишут авторы, представляющие группы исследователей в разных странах.

Близко к синему кластеру располагается фиолетовый кластер, понятия которого связаны с расширением поля информационного обучения, привлечением к компьютерным наукам не только мальчиков, но и девочек, удержанием учеников в поле компьютерных наук в состоянии потока через организацию летних школ по программированию и конкурсов, подобных «Часу кода». Неожиданным выглядит появление здесь языка программирования App Inventor, но и его присутствие в кластере понятий, расширяющих участие подростков в изучении вычислительных наук может

быть объяснено тем, что на нем легко разрабатывать приложения для мобильных устройств, чрезвычайно популярных среди молодежной аудитории.

Шестой, бирюзовый, кластер объединяет понятия, связанные со взаимодействием учащихся и коллективным созданием ими игр и моделей с привлечением больших данных и использованием машинного обучения и искусственного интеллекта. Присутствие в этом кластере обучающей среды Alice, предназначенной для создания учащимися игр в трехмерной среде, объясняется разработкой в этой среде специальных механизмов совместной деятельности, позволяющих осуществлять копирование поведения между управляемыми агентами. Например, ученик может найти в репозитории птицу, уже умеющую летать, и перенести это умение на программируемого агента в своем проекте.

Седьмой, оранжевый, кластер объединяет компьютерные вычисления, вычислительную грамотность, самооффективность и европейский конкурс по информатике Bebras.

Восьмой, коричневый, кластер дополняет темно-зеленый кластер и объединяет совместное обучение, образовательные игры, игровой дизайн, разработку игр и информатику.

Красный и фиолетовый кластеры сопряжены через связующих игроков Дениэла Гарсиа и Тиффани Барнес (García et al. 2015; García et al. 2009). И хотя на карте нет фигуры Брайна Харви, но именно он стоит за объединением этих групп, поскольку создание языка Snap! и реализованного в среде этого языка курса «Красота и радость программирования» является его заслугой.

3.2. Анализ совместной деятельности на вики-площадке Letopisi.org

Во втором параграфе представлен анализ совместной деятельности на вики-площадке Letopisi.org — общенационального образовательного проекта с международным участием, который к настоящему времени существует более пятнадцати лет. В качестве

исходного материала для анализа использовался журнал совместной деятельности участников проекта Letopisi.org, в котором были собраны все действия. Столбцы записей содержат следующие характеристики:

- Время действия.
- Имя участника.
- ID страницы.
- ID участника.
- Название страницы.
- Количество добавленных символов.

Стандартная строка в файле журнала истории выглядит следующим образом:

```
20090416061258;Н.Ефимова;135741;27831;"" город_Кунгур"";  
87.
```

Для статистической обработки данных использовался язык R и специально предназначенный для сетевого анализа пакет *igraph*. Для заключительной визуализации сетевых отношений между участниками совместной деятельности использовался пакет *VOSviewer*. Записи журнала были разделены по годам, и истории совместной деятельности за каждый год, с 2006 по 2015, доступны для дальнейшего анализа на сайте открытых данных.

Двумодальные графы были разбиты на кластеры-сообщества при помощи алгоритма *walktrap.community*. Анализ состава кластеров показал, что алгоритмическое разбиение на кластеры имеет смысл и позволяет автоматически выделить группы акторов, участвующих в совместной деятельности.

Дальнейший анализ взаимосвязей заключался в преобразовании (проекции) двумодального графа в одномодальные графы авторов и страниц. Для узлов одномодальных графов также определялись сетевые показатели и проводилось разбиение сети на кластеры. Заключительный анализ и визуализация кластеров-сообществ участников совместной деятельности осуществлялись при помощи пакета *VOSviewer*. При этом использовался режим построения карты на основании данных, в которых указываются

связи узлов и сила этих связей. Алгоритм определения сообщества и выбор способа представления графа учитывали силу ассоциаций между узлами. Параметры притяжения и отталкивания были подобраны так, чтобы обеспечить максимальную видимость кластеров, и соответствовали +3 и -1. Карта истории совместной деятельности в социотехнической системе Letopisi.org в 2006 году представлена на рисунке 39.

3.3. Анализ совместной деятельности учителей нескольких школ внутри их школьных доменов Google Apps

В третьем параграфе представлен анализ совместной деятельности учителей нескольких школ внутри их школьных доменов Google Apps. В качестве источника данных, которые использовались для анализа в исследовании, мы взяли записи о действиях с документами в трех доменах Google Apps для образования. Для автоматического извлечения записей была написана отдельная программа на языке сценариев Google Script, использующая возможности Google Reports API. Необходимо отметить, что таким образом можно извлекать из домена данные за последние шесть месяцев. Если перед исследователем встает задача получения картины за более длительный период (скажем, за год), то следует использовать какой-либо специализированный инструмент, например General Audit Tool Core, доступный в качестве приложения для домена Google Apps. В нашем случае после извлечения данных из домена мы получили их в виде таблицы приблизительно в 90 000 записей. Каждая запись при этом представляет собой информацию о действии, совершенном пользователем (участником, зарегистрированным в домене), по отношению к объекту, хранящемуся в домене, имеющему конкретный идентификационный номер и конкретного автора. Были извлечены данные за первое полугодие 2015 года. При анализе использовались два типа действий субъекта: создание и редактирование.

На следующем рисунке представлена позиция, которая сложилась на компьютерной карте совместной деятельности внутри

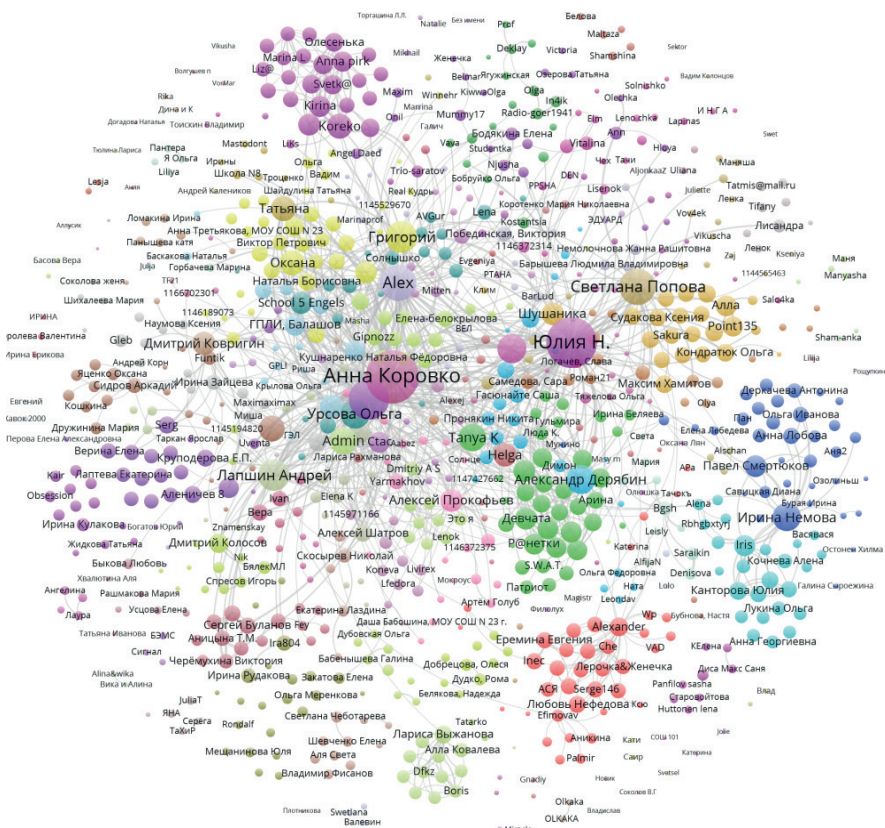


Рис. 39. Карта сообществ Letopisi.org (2006 г.)

школьной информационной системы после четырех тысяч действий, совершенных участниками (рис. 40).

Диаграмма может быть использована для анализа отношений, которые складываются между участниками совместной деятельности. Например, на социограмме около поля Р9 находится документ, созданный учителем английского языка. Этот документ привлек внимание учителей начальных классов и сотрудника-психолога. В другой части поля совместной деятельности Н10 презентация учителя информатики привлекла внимание учителей математики. Модель позволяет увидеть множество просмотров и несколько соредактирований данного объекта совместной сетевой деятельности.

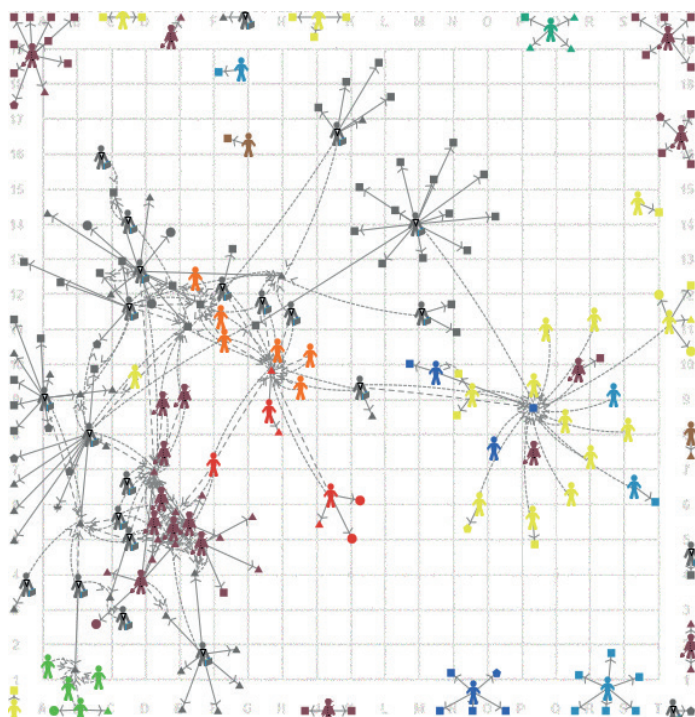


Рис. 40. Поле совместной деятельности
в информационной системе школы

3.4. Анализ совместной деятельности учеников в одной из студий на сайте Scrtatch

В четвертой части третьей главы представлен анализ совместной деятельности учеников в одной из студий на сайте Scrtatch: <https://scratch.mit.edu/studios/4789981/>

В 2018 году Московский городской педагогический университет и Национальное общество технологий в образовании объединили усилия с командой Всемирного банка в рамках российской сети для проведения экспериментов с новыми видами учебной деятельности, формирующими навыки XXI века. Стоит отметить, что организаторы сети обладали многолетним опытом

организации общественных и образовательных проектов на стыке образовательной политики и учебной деятельности. Этот проект был нацелен на изучение цифровой грамотности в сфере всех навыков XXI века. В рамках цифровой грамотности акцент был сделан на навыки совместной работы. Организаторы решили найти и разработать учебную деятельность, которая поможет сформировать способность и желание учащихся делиться результатами своей деятельности, работать в команде и распределять задания. Они выбрали Scratch как среду, в которой уже есть инструменты для организации и мониторинга такой деятельности. В Москве и других городах России прошли хакатоны, в ходе которых ученики осваивали новые виды учебной деятельности, направленные на формирование навыков совместной сетевой деятельности. Формат хакатонов был выбран как наиболее соответствующий поиску инновационных решений. Всего было проведено более 90 хакатонов, в которых приняли участие более 4000 школьников. В рамках хакатонов внимание уделялось навыкам и инструментам совместной деятельности. В ходе тестирования учителя смогли определить недостатки и неопределенности предлагаемой учебной деятельности и внести изменения в шаблон. Эта работа проводилась большей частью на местах — в школах и центрах дополнительного образования. Наибольшее внимание в ходе семинаров и хакатонов уделялось использованию инструментов совместной деятельности. На финальном этапе конкурса жюри оценивало не только сам окончательный проект, но и материалы студии, деревья ремиксов, т. е. то, насколько деятельность членов команды соответствовала разработанному сценарию учебной деятельности. Мы провели анализ взаимоотношений участников нескольких скретч-студий, опираясь на действия, связанные с публикацией и комментированием проектов. На рисунке 41 представлена социограмма отношений внутри студии Scratch. Фигурки людей обозначают участников/скретчеров, страницы кода — проекты. Сплошная линия от скретчера к проекту означает, что этот скретчер является автором проекта. Пунктирная линия означает, что скретчер комментировал проект.

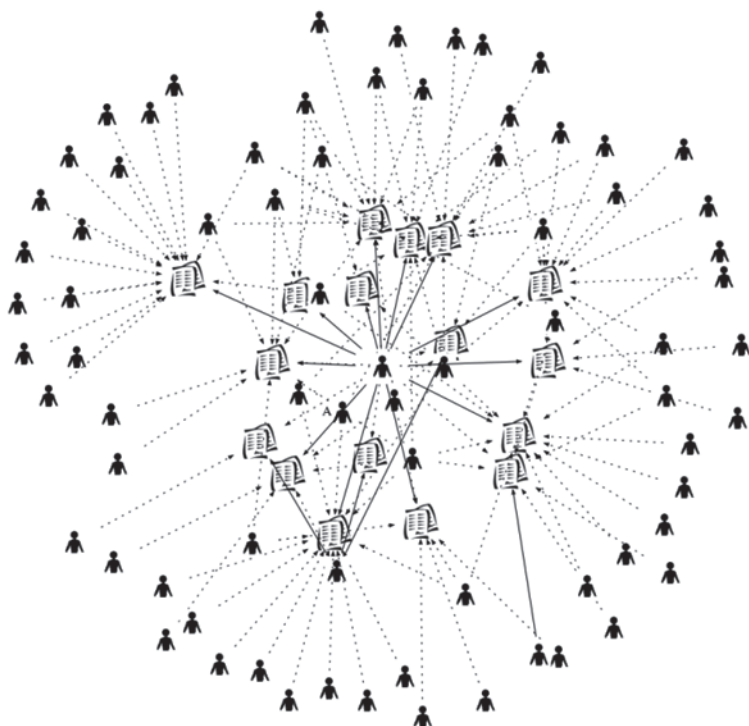


Рис. 41. Социограмма авторов и проектов в студии Scratch

Материалы студии, в которой происходил сбор проектов к конкурсы открыты. Кроме того, у нас есть доступ к материалам этой студии через Scratch API: <https://api.scratch.mit.edu/studios/4789981>

Мы использовали возможность обратиться ко всем записям о проектах этой студии и получили их перечень с указанием авторов и описанием проектов, благодаря использованию запросов: <https://api.scratch.mit.edu/studios/4789981/projects/?limit=40&offset=0>

Мы можем обращаться к материалам студии Scratch из проектов Snap! и конвертировать JSON файлы в понятные читателям таблицы (см. рис. 42).



Рис. 42. Превращение JSON файла в таблицу SNAP!

Кроме того, что мы можем работать с данными студии о проектах, мы можем работать с данными отдельных участников и устанавливать между ними связи на основании того, кто из них кого отслеживает. Пример извлечения такой информации представлен на рисунке 43.

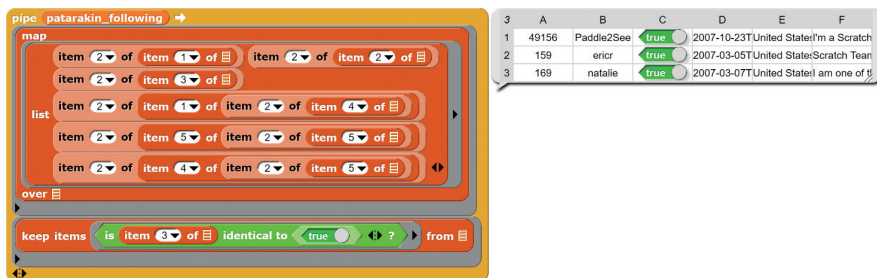


Рис. 43. Фильтрация членов команды из списка отслеживаемых участников

3.5. Совместная деятельность участников МЭШ по созданию и повторному использованию сценариев уроков

В последней части третьей главы мы исследуем совместную деятельность участников московской электронной школы по созданию и повторному использованию сценариев уроков. Принципы анализа совместной деятельности внутри этого образовательного репозитория описаны в статье «Незримый колледж МЭШ» (Патаркин, Буров 2022). В рамках данной главы мы представим процедуру выделения участников из фракции реципрокного копирования сценариев уроков. Для обработки данных, их визуализации и проведения статистического анализа мы использовали возможности языка R. При этом наибольшее внимание было уделено подходу аккуратного представления данных, который делает R современным и понятным средством, позволяющим совершать точные операции над таблицами и графами (Baumer et al. 2017; Wickham 2015; Wickham, Grolemund 2016; Wild 2016).

Исходно датасет содержал данные о действиях 64 тыс. учителей, которые сотрудничают внутри репозитория цифровых объектов Московской электронной школы. В приведенном примере использованы данные, которые были доступны в системе на май 2020 года. Поскольку у нас есть цифровой след действий над цифровыми объектами совместного использования, мы можем построить электронную карту совместных действий участников в системе МЭШ, основанную на истории отношений исследуемого участника с другими участниками совместной деятельности. Мы можем связать участников между собой, используя их действия над созданными объектами. Логика таких связей была следующей:

- субъект (агент деятельности А) создает объект, который в дальнейшем используется другими субъектами деятельности;
- если агент Б копирует объект, созданный агентом А, то возникает связь от агента Б к агенту А, и это связь действия типа `remix`;
- если агент Б запускает объект агента А, то возникает направленная связь от агента Б к агенту А, и это связь действия типа `launch`;

- если агент Б вставляет объект агента А в домашние работы, то возникает связь от агента Б к агенту А, и это связь действия типа `homework`;
- если агент Б добавляет объект агента А в избранное, то возникает направленная связь от агента Б к агенту А, и это связь типа `favorited`.

Мы включили в обобщенную таблицу индивидуальных показателей всех участников различные показатели их центрально-сти, локальный коэффициент кластеризации и количество троек (`triangle`), в которые входит каждый учитель.

Процедура фильтрации представителей этой фракции включала несколько этапов. В начале мы выделили только тех, кто создавал собственные сценарии, поскольку если таких созданных сценариев нет, то к участнику не могут быть направлены внешние связи.

Функция выделения записей участников — создателей сценариев:

```
creative_table <- function(p) {
  cu <- p %>%
    filter(action == "create") %>%
    select(user) %>%
    distinct() %>%
    pull()
  ctable <- p %>%
    filter (user %in% cu)
  return(ctable)
}
```

Отфильтрованные записи были преобразованы в одномодальный граф, объединяющий только участников, создававших учебные сценарии. Из 8031 узла этого графа были отобраны только те, у кого есть взаимные связи через взаимное копирование учебных сценариев. Все узлы, которые при такой фильтрации утратили связи и превратились в изолированные, были удалены.

```
cgraph %>%
+ activate(edges) %>%
+ filter(action == "Remix") %>%
```

```
+ filter(edge_is_mutual()) %>%
+ activate(nodes) %>%
+ filter(!node_is_isolated())
```

В результате мы получили граф из 147 отдельных компонентов

```
# A tbl_graph: 1431 nodes and 18855 edges
#
# A directed multigraph with 147 components
```

Следующий важный этап состоял в переводе таблицы действий участников в граф отношений между участниками — создателями сценариев. Принципиальный момент выделения среди всех связей только тех, которые взаимны. Мы рассматриваем связи между участниками как основание для связи, только если такие связи взаимны. В МЭШ взаимная связь образуется в том случае, если учитель А скопировал сценарий урока, созданный учителем Б, а учитель Б скопировал сценарий урока, созданный учителем Б. После удаления из системы однонаправленных не взаимных связей в системе остается множество одиноких участников, которые не входят ни в какие группы и могут быть удалены. Граф, в котором собраны все участники, взаимно копирующие учебные сценарии друг у друга описывается следующими параметрами:

```
# A tbl_graph: 1552 nodes and 21349 edges
# A directed multigraph with 145 components
```

В этом направленном графе 1552 узла, объединенных между собой 21 349 связями. Граф совместных копирований состоит из 145 компонент. При этом узлы и связи распределены крайне неравномерно. Главный компонент содержит 1174 узла и 17 928 связей, а на остальные 144 компоненты приходится всего 378 узлов и 3421 связь.

Визуализация графа (рис. 44) получена путем применения следующих фильтров и слоев:

```
rcgraphR %>%
  activate(edges) %>%
  filter(!edge_is_multiple()) %>%
  convert(to_undirected) %>%
  activate(nodes) %>%
```

```

mutate(c_betw = centrality_betweenness()) %>%
  activate(edges) %>%
  mutate(edge_between = centrality_edge_betweenness()) %>%
  activate(nodes) %>%
  ggraph(layout = 'stress') +
  geom_node_point(aes(size = c_betw, colour = c_betw,
    shape = '21')) +
  geom_edge_link0(aes(alpha = edge_between), edge_
    colour = "grey66") +
  theme(legend.position = 'none')

```

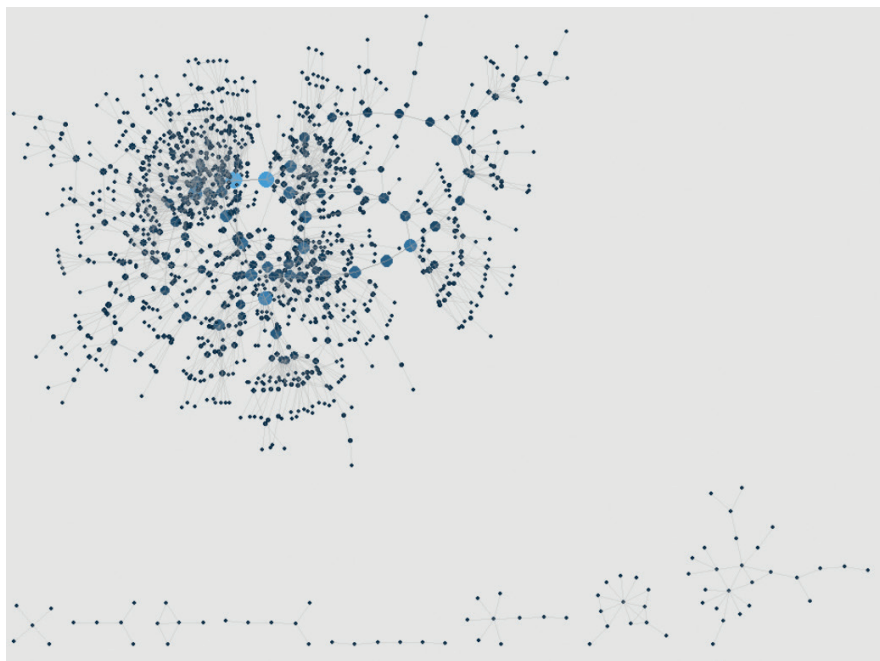


Рис. 44. Граф взаимных копирований

ГЛАВА 4

МОДЕЛИРОВАНИЕ СОВМЕСТНОЙ ДЕЯТЕЛЬНОСТИ

В главе представлены модели, которые использовались для описания и понимания совместной деятельности, происходящей на базе сетевых сообществ. При этом при разработке моделей мы стремились выращивать в искусственных сообществах феномены, которые наблюдали и описывали в третьей главе данной монографии.

Разработка многоагентной модели и ее использование предполагает решение следующих задач:

- анализ действий участников сетевых сообществ по созданию и использованию цифрового образовательного контента: текст, видео, цифровые истории, модели, сценарии и презентации образовательных практик и т. д.;
- построение обобщенной многоагентной модели деятельности по созданию и использованию цифрового контента в сетевых сообществах;
- изучение агентов, имитирующих различные типы поведения внутри образовательных сообществ, на базе модели управляющих воздействий на искусственные сообщества;
- разработка методов управления деятельностью участников образовательных сообществ, основанная на результатах управления поведением агентов в искусственном сообществе.

Цифровая среда, в которой организуется современная совместная деятельность, позволяет сохранять данные о действиях субъектов совместной деятельности и на основании этих данных выстраивать связи, которые возникают между субъектами и объектами деятельности. Действия, которые совершают акторы в цифровой среде, очень похожи на процедуры, которые выполняют черепашки в известной модели «Термиты» (Resnick 1997), когда каждая черепашка следует простым правилам.

Если ты идешь и ничего не несешь и встречаешь на своем пути деревянную палочку (chip), то возьми ее. Если ты идешь

с деревянной палочкой и встречаешь другую палочку, то положи щепку, которую несешь, и двигайся дальше.

Каждая черепашка выполняет последовательность процедур: `search-for-chip` – `find-new-pile` – `put-down-chip`, — и в результате из множества случайным образом разбросанных по экрану палочек собирается одна круглая куча.

Модель термитов кажется наиболее привлекательной, поскольку в ней присутствуют палочки — `chips` — как объекты совместной деятельности. Экспериментируя с этой моделью, мы можем глубже понять феномены совместной деятельности. Представим себе, что у термитов есть журнал учета рабочих действий, куда они записывают все свои действия со щепочками. Таким образом, если участник совершает какое-то результативное действие с `chip`, то он об этом действии оставляет запись в журнале. Для того чтобы проверить, что дают нам попутные записи в журнал, мы несколько видоизменили исходный текст модели «Термиты», добавив к ней новые переменные и правила. В модели появилась переменная список `WIKILOG`, куда термиты записывают отчеты о своих действиях. В процедурах `search-for-chip` и `put-down-chip` были сделаны небольшие добавления. В модель была добавлена процедура, которая на основании записей в журнале устанавливает связь между агентами, которые перетаскивали одну и ту же палочку (Patarakin 2017).

Эта простейшая модель создания социограмм на основании действий с социальными объектами может быть применена к различным ситуациям совместной деятельности. Во всех случаях, когда агенты деятельности производят действия над объектами совместной деятельности, журнал этих действий может служить исходным материалом для построения социограмм и проведения социального исследования. Например, мы можем смоделировать ситуацию, когда 1000 независимых агентов в течение 30 000 ходов переносят 6000 палочек. Модель реализована на языке NetLogo 6.1 и исходный код доступен в сети Интернет: http://modelingcommons.org/browse/one_model/5071

Интерфейс видоизмененной модели представлен на рисунке 45. Кнопка `Logs_to_sociogram` убирает с экрана все палочки и показывает связи между агентами, которые перемещали одни и те же палочки.

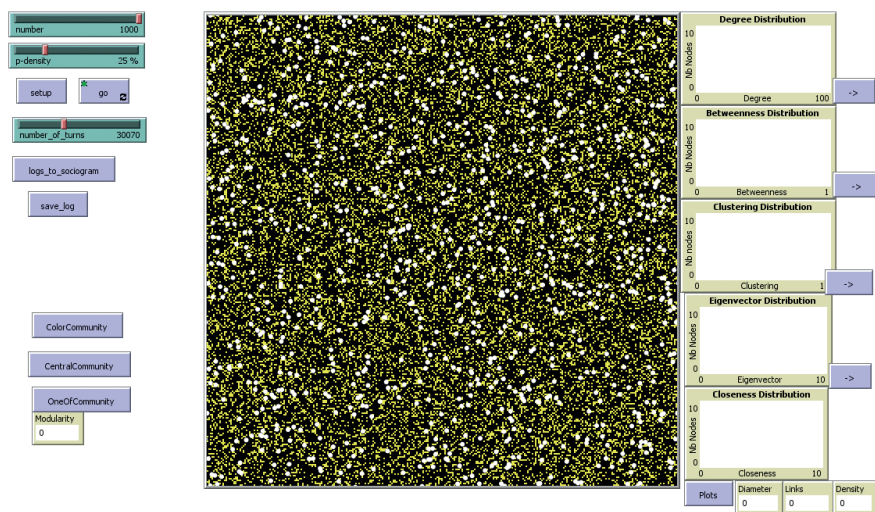


Рис. 45. Интерфейс модели «Термиты с журналом»

После того как все действия были совершены и записаны в журнал, модель убирает с экрана палочки и показывает связи между термитами (рис. 46).

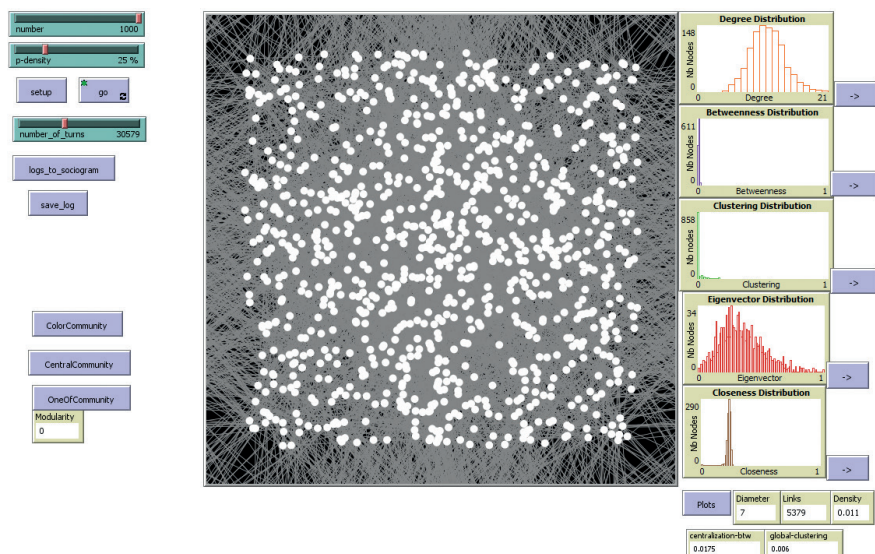


Рис. 46. Связи на основании действий термитов

В правой части экрана на рисунке 46 представлены гистограммы распределения акторов по сетевым характеристикам:

- центральность по степени (Degree centrality) — показывает, насколько много связей у данного узла;
- центральность по посредничеству (Betweenness) — показывает вероятность того, что данный узел контролирует информационные потоки внутри системы совместной сетевой деятельности;
- центральность по собственному вектору (Eigenvector centrality) — показывает насколько данный узел связан с другими важными узлами;
- центральность по близости (Closeness centrality) — показывает, насколько быстро можно добраться от данного узла до других узлов сети;
- коэффициент кластеризации (Clustering coefficient) — показывает, насколько соединены между собой узлы, связанные с данным узлом. Локальный коэффициент кластеризации достигает высокого уровня у тех узлов, которые включены в состав групп.

Дальнейший анализ и визуализация данных журнала об истории действий термитов позволяют увидеть цветные кластеры акторов, соединенных между собой большим числом связей, чем с другими акторами (рис. 47).

Модель «Термиты с журналом» позволяет увидеть распределение сетевых характеристик в системе, когда участники совместной деятельности действуют вслепую и не обращают друг на друга никакого внимания. Это крайний случай совместной сетевой деятельности, с которым мы можем сравнивать ситуации, которые складываются в реальных сетевых образовательных сообществах, где роль палочек играют статьи, модели, цифровые истории или фрагменты нормативных документов, которые создают, обсуждают или оценивают субъекты совместной деятельности.

В качестве исходного материала для анализа использовался журнал совместной деятельности участников проекта Letopisi.org, в котором были собраны все действия за десятилетний период с 2006 по 2015 год. Записи журнала были разделены по годам,

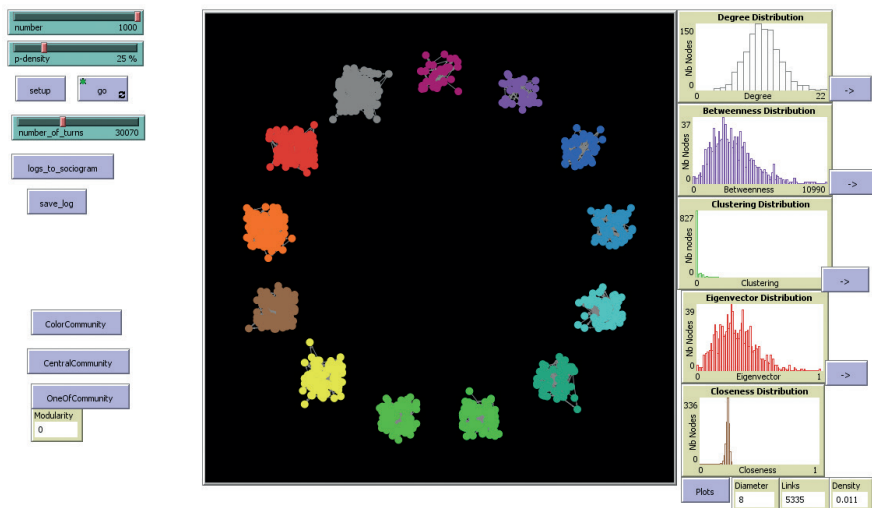


Рис. 47. Кластеры термитов, связанных общими объектами

и истории совместной деятельности за каждый год были доступны для анализа. Анализ взаимосвязей заключался в преобразовании (проекции) двумодального графа в одномодальные графы авторов и страниц. Для узлов одномодальных графов определялись сетевые показатели и проводилось разбиение сети на кластеры. Сетевые характеристика акторов сетевого сообщества и один из кластеров участников представлены на рисунке 48. По сравнению с моделью «Термиты с журналом» мы видим гораздо более неравномерное распределение сетевых характеристик между участниками совместной деятельности.

Для объяснения и генерации феноменов, которые наблюдаются в репозитории сценариев уроков МЭШ, мы использовали модели образования стаи и формирования команд. Например, понятие гигантской компоненты внутри МЭШ получает конкретное численное выражение, поскольку мы установили, что 75 % учителей, копирующих проекты друг друга на принципах взаимности, включены в состав гигантской компоненты.

Мы использовали модель Flocking для генерирования и анализа данных о взаимосвязи независимой переменной «область

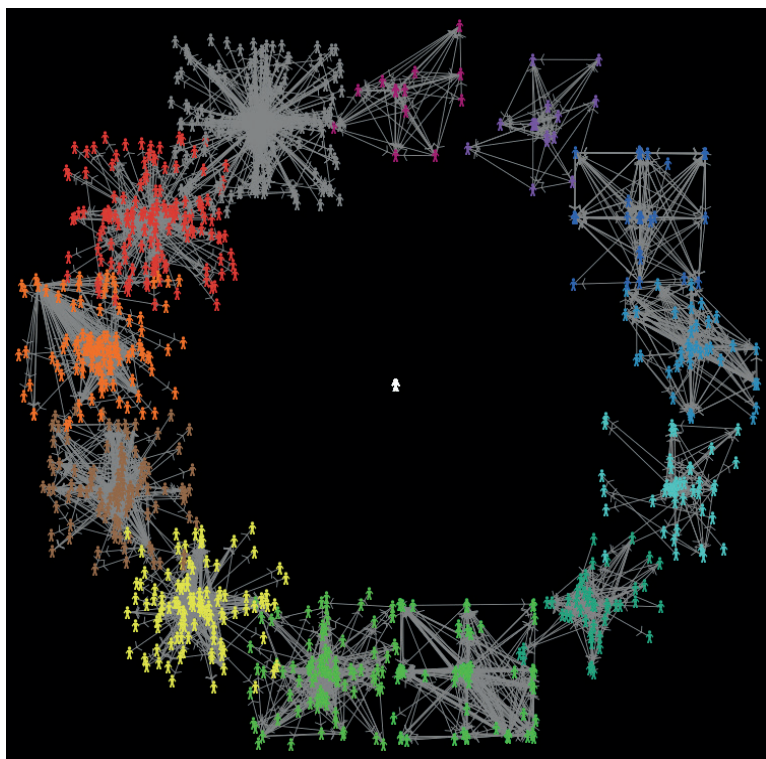


Рис. 48. Цветные кластеры проекта Letopisi.org

обзора» с формированием общего направления, в котором движется стая агентов. При этом мы предположили, что параметр «область обзора» может быть использован в качестве инструмента управления или воздействия на поведение участников. В данном случае мы рассматриваем расстояние, на котором агент отслеживает своих партнеров по стае, не как степень подверженности человека воздействию со стороны своих партнеров, а как возможность, которую ему предоставляет среда. Например, внутри репозитория МЭШ дизайнеры могут приложить отдельные усилия для того, чтобы участник видел как можно больше сценариев уроков, созданных другими учителями. Чтобы проанализировать, как увеличение параметра «область обзора» повлияет на общее направление стаи и формирование коллективного поведения

участников МЭШ, мы поставили эксперимент со следующими условиями записанными в BehaviorSpace:

```
["max-cohere-turn" 2]
["max-separate-turn" 2]
["vision" 1 2 3 4 5]
["minimum-separation" 1]
["population" 400]
["max-align-turn" 4]
```

В качестве переменной изменялось значение параметра «область обзора». Как измеряемая величина в десяти прогонах каждой серии эксперимента фиксировалось отклонение от средней направленности общего движения стаи агентов. Результаты в виде графика boxplot представлены на рисунке 49, на основании которого можно сделать заключение, что увеличение зоны видения серьезно сказывается на общей направленности коллективного поведения в очень узком интервале значений. После этих значений увеличение видимости не приводит к каким-либо серьезным изменениям в поведении системы.

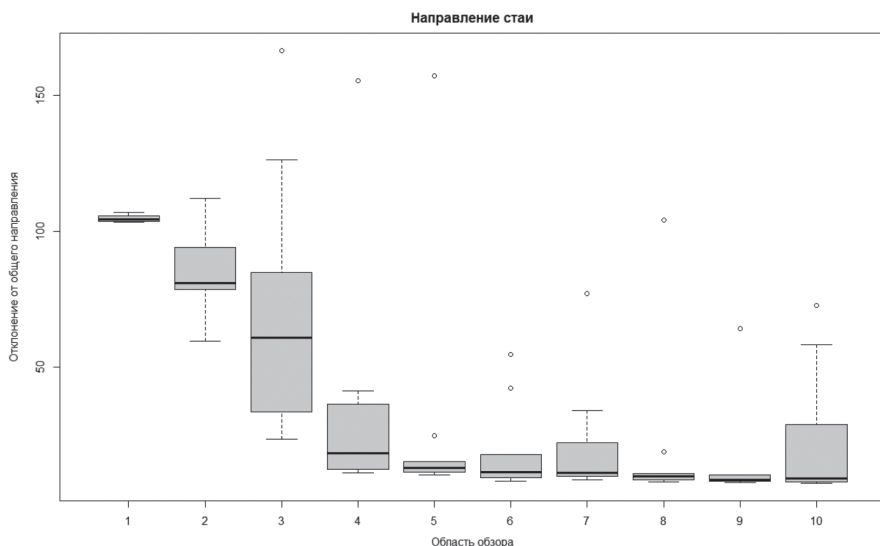


Рис. 49. Взаимосвязь области обзора с направленностью стаи

Для того, чтобы объяснить объединение 75 % учителей, копирующих друг у друга сценарии уроков, в составе гигантской компоненты, мы использовали модель формирования команд, внося в нее изменения, отражающие специфику репозитория Московской электронной школы. Во-первых, мы снизили минимальное число участников, необходимое для образования команды, с трех до двух, поскольку в МЭШ часто наблюдаются группы-диады. Во-вторых, мы увеличили время пребывания агента в активном состоянии до того, как он покинет рабочую среду в силу своего возраста, поскольку сама система МЭШ пока еще очень молода, и этот параметр можно не использовать. После этих изменений мы поставили эксперимент со следующими условиями, записанными в BehaviorSpace:

```
["max-downtime" 800]  
["team-size" 2]  
["p" [10 5 75]]  
["q" 52]
```

В качестве независимой переменной использовалось значение p — вероятность выбора новичков в качестве партнера по команде опытного участника, уже входящего в состав группы. В качестве измеряемой величины в десяти прогонах каждой серии эксперимента фиксировался процент участников, входящих в состав гигантской компоненты, после 800 циклов модели. Результаты в виде графика `boxplot` представлены на рисунке 50, на основании которого можно сделать предварительные выводы о том, что заметное изменение воздействия величины p на величину гигантской компоненты происходит в интервале 35–45 %, и выявленное для МЭШ объединение 75 % учителей может быть объяснено тем, что вероятность выбора опытного участника, уже входящего в состав группы, составляет не менее 60 %.

В своей модели в среде NetLogo мы намерены были получить данные и визуализации, которые помогут нам обсудить и лучше понять события и сетевые структуры, возникающие в результате взаимодействия множества агентов в среде Московской электронной школы.

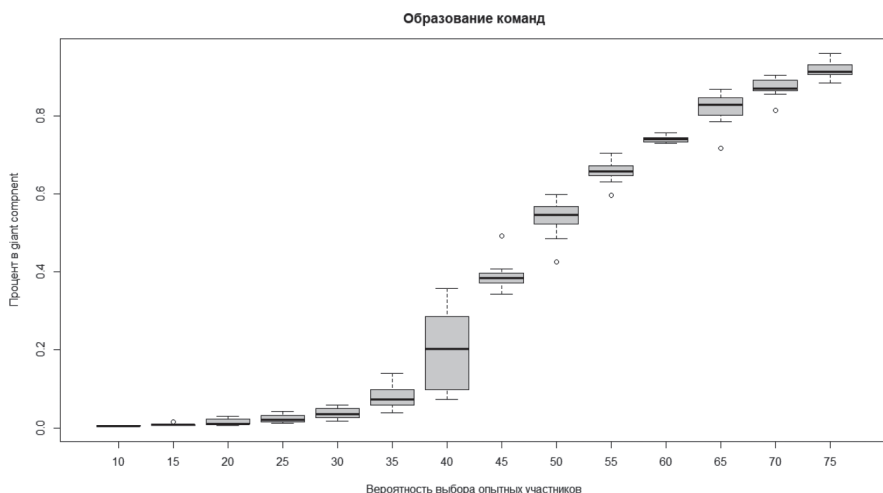


Рис. 50. Связь вероятности выбора опытного участника и величины гигантской компоненты внутри МЭШ

Искусственное сообщество, имитируемое многоагентной моделью, устроено следующим образом. В созданном в рамках многоагентной модели искусственном мире есть поле, по которому могут перемещаться агенты-учителя и агенты-сценарии учебной деятельности.

Многоагентная модель в среде NetLogo была разработана нами специально для анализа совместной деятельности внутри системы Московской электронной школы и подробно описана в статье (Patarakin et al. 2021). Модель специально разрабатывалась для имитации ситуации создания и использования контента в цифровой среде, где действуют множество агентов-учителей, связанных со школами, в которых они преподают и с учебными дисциплинами, которыми они размечают создаваемые в системе цифровые объекты.

```
users-own [
;; свойства всех участников - учителей
agentname ;; ID агента
School ;; школа, где он преподает
Age ;; Возраст (не использован для разделения)
```

```

Gender ;; пол агента — выбираются по тому, насколько
они близки
User_Subj ;; перечень учебных дисциплин
Action ;; действия, которые совершил
t-pagenum ;; имя созданной страницы
region_point ;; место рождения
]

```

Исходно в модели использована метафора мяча как социального объекта, относительно которого выстраивается взаимодействие участников. Поведение агентов в модели регулируется следующими правилами: агент-учитель определяет, есть ли в его окружении потенциальные сценарии, к которым он мог бы прикоснуться. Если такие объекты есть, то агент выбирает ближайший и делает к нему шаг:

```

to go
;; if length reused > length created [stop]
;; ask users with [ (any? balls-here )] [touch_ball
one-of balls-here ]
ask users with [not any? balls-here ] [ search_ball ]
ask balls [
ball_move
]

```

Если учитель касается сценария, то он совершает действие над этим объектом. Вариант действия зависит от того, касался ли кто-нибудь прежде этого объекта. Если это первое соприкосновение, то действие учителя заключается в создании сценария. При создании сценария учитель «навешивает» на этот сценарий метку-ярлык одной из тех учебных дисциплин, которые он преподает. В результате этого действия объект меняет свое положение, а в журнале появляется запись, что агент-учитель по отношению к агенту-сценарию совершил действие-создание, и это создание произошло в определенный момент времени:

```

let player one-of users-here
let newpage 1 + length pages
set pages lput newpage pages
set pagenum newpage ;;
set Ball_Subj one-of [User_Subj] of player ;;

```

```
set Gender [Gender] of player ;;  
set wikilog lput ( se [who] of player pagenum "create"  
[School] of player Ball_Subj ) wikilog  
set heading [heading] of player  
set color green
```

Учителя исходно связаны со школами — определенными географическими точками на поверхности. При запуске модели сначала задаются школы, а потом вокруг них возникают учителя. В рамках данной модели существует ряд ограничений. Одно из таких ограничений состоит в том, что учителя не могут изменять свою принадлежность к школе. При возникновении учителя в системе, кроме принадлежности к школе, учителю присваивается связь с одной или несколькими учебными дисциплинами, название которых учитель может в дальнейшем использовать при создании учебных сценариев.

В действующей модели последовательность событий определяется последовательностью записей в журнале. NetLogo позволяет устанавливать временные метки на каждое действие, и одно из направлений развития моделей будет связано с изучением динамики поведения агентов.

Если мячик-сценарий уже был одним из учителей открыт, то это уже авторский сценарий, и агент-учитель совершает над ним действие по повторному использованию. В результате этого действия мяч меняет свое положение на экране, а главное, в школьном журнале появляется запись, что агент-учитель по отношению к агенту-мячику совершил действие — повторное использование, и это повторное использование произошло в определенный момент времени.

Варианты повторного использования могут быть заданы в системе таким образом, чтобы соответствовать действиям, которые допустимы над учебными сценариями в среде репозитория Московской электронной школы: просмотр, добавление в избранное, запуск сценария, скачивание, использование в домашних заданиях.

Вероятность выбора одного из таких действий может зависеть от их сложности или от предрасположенности учителей к совершению

тех или иных действий. В этом плане вполне допустимым действием агента над объектом может стать отсутствие действия. При этом отсутствие действия над объектом никак не фиксируется в электронном журнале. В настоящее время отсутствие действия учителя над объектом реализуется в одном из вариантов поведения модели, когда существует разделение учителей и объектов по учебным дисциплинам. В этом варианте модели при встрече учителя с объектом, отмеченным ярлыком дисциплины, отсутствующей в перечне дисциплин, которые преподает данный учитель, агент-учитель не производит над объектом никаких действий.

В ходе деятельности агентов-учителей над агентами-сценариями побочные результаты этой деятельности отображаются на гистограмме распределения свойств сценариев. На гистограмме показано число потенциальных сценариев, число созданных и число повторно использованных сценариев.

После того как участники совершат над объектами деятельности заданное исследователем число действий, активность системы останавливается и исследователь может получить граф отношений, которые сложились между участниками в ходе их совместной деятельности над объектами. В системе действуют следующие правила образования связей между агентами: если агент А повторно использовал объект, созданный агентом Б, то между А и Б возникает направленная от А к Б связь, показывающая, что А использовал Б. На практике А использовал объект, созданный Б, но в этой связи мы можем не учитывать объект. При этом мы можем учитывать в свойствах связи свойства принадлежности к определенной дисциплине.

В системе действует несколько параметров, которые определяют поведение участников и которыми исследователь может управлять.

Ранее мы уже упоминали о возможности выбора действий по повторному использованию сценария в зависимости от выбора исследователем положения рычажка «Разделение по предметам».

Кроме этого, мы можем определять дальность зоны интересов: чем больше значение этого параметра, тем дальше вокруг себя

видит учитель и тем в большем диапазоне вокруг себя он будет искать мячики-сценарии. Возможно, что на будущее нужно предусмотреть, что учитель в своем диапазоне поиска ищет исключительно новые сценарии, с которыми он ранее еще не взаимодействовал.

Зона принадлежности к школе — рычажок, ограничивающий перемещение учителя в сторону заинтересовавших его объектов. Если этот сценарий выходит за зону принадлежности к школе, то учитель не делает шага в его сторону.

Сдвиг объекта в результате воздействия на него учителем. Параметр регулирует то, насколько далеко перемещается объект-сценарий в результате взаимодействия с учителем. Смысл параметра заключается в том, что действие учителя над объектом может изменять положение этого объекта относительно других учителей. В результате к объекту могут получить доступ учителя из других школ.

При выработке принципов построения многоагентной модели деятельности по созданию и использованию цифрового контента в сетевых сообществах предполагается, что в системе изначально присутствуют два типа агентов — субъекты деятельности и объекты деятельности.

Типы агентов в средах многоагентного моделирования определяются как породы (breeds).

```
breed [учителя учитель] ;; порода участников — авторы  
и редакторы сценариев  
breed [сценарии сценарий] ;; порода сценариев — сцена-  
рий создается учителем
```

Каждая порода агентов изначально получает набор характеристик — свойств, которыми будут обладать особи (элементы) этой породы. Мы можем заранее вне зависимости от конкретной среды реализации модели определить свойства особей каждой породы.

Характеристики субъектов деятельности:

- Свойство «Имя агента» (Agentname) — ID участника, по которому к нему можно обратиться и идентифицировать. Уникальный номер учителя возникает в момент его создания в системе.

- Свойство «Школа» (School) — школа, в которой работает учитель. При создании экземпляра каждого учителя он получает свойство — школу, в которой работает. Позднее мы выбрали вариант, когда на поле деятельности возникают образовательные центры — школы, и эти школы порождают учителей. В системе могут быть приняты правила, разрешающие смену свойства. По умолчанию свойство остается неизменным и позволяет проследивать формирование внутришкольных групп учителей.

- Свойство «Возраст» (Age) — присваивается учителю при создании его как экземпляра. Это свойство изменяется по циклам совершения действий. При каждом цикле в системе к возрасту добавляется +1. При достижении определенного возраста активность агента по отношению к объектам деятельности сокращается. Возможный вариант использования свойства Age — снижение с возрастом скорости перемещения агента в системе.

- Свойство «Пол» (Gender, F/M) — присваивается учителю при создании его как экземпляра. Изначально неизвестно, как свойство влияет на деятельность, но можно предположить, что объекты могут разделяться и фильтроваться при показе в зависимости от пола того, кто их создал.

- Свойство «Психотип» (Psychotype, pType) — мы можем выбрать из известных типологий (шизоид, истероид, эпилептоид, гипертим и астеник) или воспользоваться типологией компьютерных игр Бартла, или оставить их по номерам: pType1, pType2, pType3, pType4, pType5, pType6, pType7. Свойство «Психотип» может влиять на активность перемещения агентов в системе. Кроме того, объекты могут разделяться и фильтроваться при показе в зависимости от психотипа того, кто их создал.

- Свойство «Стаж трудовой деятельности» имеет, на наш взгляд, значение именно по отношению к деятельности внутри системы и должно учитывать действия участника внутри системы. Эти динамические характеристики могут извлекаться из истории действий, которые мы собираем для всех участников внутри системы.

- Свойство «Учебная дисциплина» — при создании элемента агента «учитель» ему в качестве этого свойства присваивается список дисциплин (User_Subj), которые он преподает. При создании объекта «сценарий» учитель присваивает ему свойство «Учебная дисциплина» — один из элементов списка тех дисциплин, с которыми он работает.

- Свойство «Уровень образования» — присваивается учителю в момент его появления в системе. Позднее при создании объектов учитель передает им свое свойство «Уровень образования».

Фрагмент интерфейса модели с управляющими кнопками и рычажками представлен на рисунке 51.

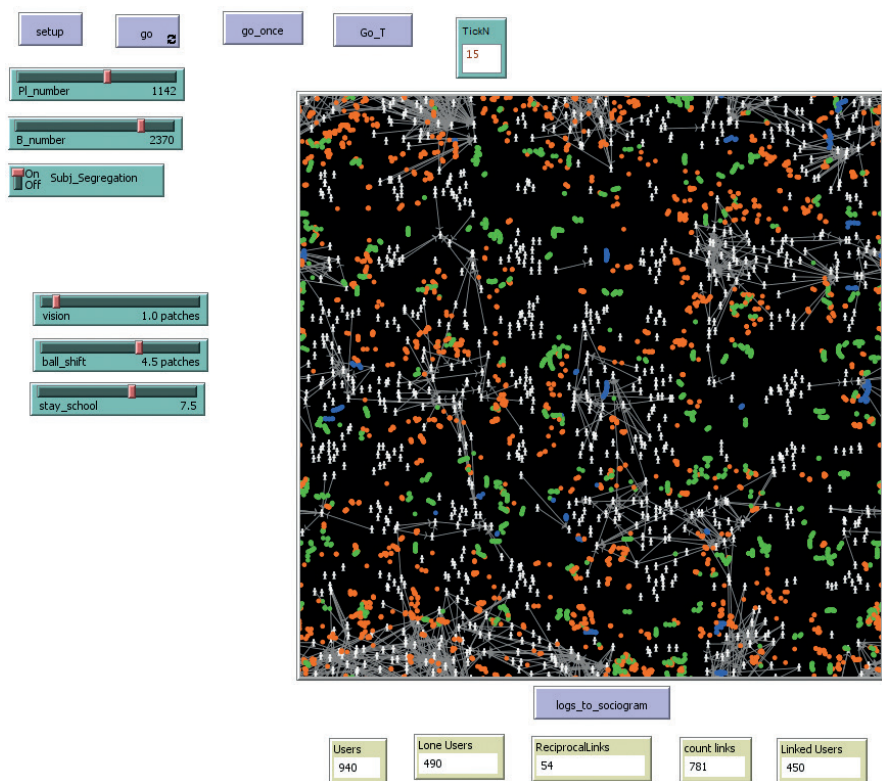


Рис. 51. Фрагмент интерфейса модели

В центре экрана расположена рабочая область, где действуют агенты двух типов : учителя (человечки) и учебные сценарии (мячики).

Кнопка *setup* очищает экран, запускает обновление всех показателей модели, расставляет по полю центры школ, с которыми соотносят свое положение учителя, расставляет по полю потенциальные сценарии уроков, которые могут быть обнаружены (открыты и созданы) агентами-учителями.

Под кнопкой *setup* находятся рычажки, при помощи которых исследователь может управлять исходным числом агентов, принимающих участие в совместной деятельности.

Первый рычажок *Pl_number* задает максимальное число агентов-учителей, которые могут принимать участие в имитации деятельности.

Второй рычажок *B_number* определяет число агентов-сценариев, которые разбросаны по полю и с которыми агенты-учителя потенциально могут встретиться (создать).

Группа рычажков *Subj_Segregation*, *vision*, *ball_shift*, *stay_school* задает возможность управляющих воздействий через ограничения, которые исследователь может накладывать на поведение агентов в искусственном сообществе.

Переключатель *Subj_Segregation* / «Разделение по предметам» определяет два возможных варианта поведения учителей при их встрече с объектами-сценариями. В случае если установлено положение *Off*, агент-учитель при встрече с объектом, который уже был создан другим агентом-учителем, совершает над ним повторное действие. Во втором случае, если установлено положение *On*, при встрече учителя с объектом, отмеченным ярлыком дисциплины, которого нет в перечне дисциплин, преподаваемых данным учителем, агент-учитель не производит над объектом никаких действий.

Рычажок *vision* определяет то, как далеко вокруг себя видит агент: ограничивается ли поле его зрения объектами, которые стоят совсем близко, или он может обратить свое внимание на объект, расположенный дальше.

Рычажок `ball_shift` определяет изменение позиции объекта сценария, после создания или повторного использования его другими участниками: останется ли объект на том же месте или он выпадет из поля зрения создателя и попадет к другим учителям (возможно из другой школы).

Рычажок `stay_school` определяет, как далеко агент может уходить от своей школы по полю совместной деятельности в поисках новых цифровых объектов.

Кнопка `go` запускает безостановочное перемещение агентов-учителей по экрану и их действие над объектами-сценариями.

При нажатии кнопки `go_onse` учителя и объекты совершают один такт общего действия.

При нажатии кнопки `Go_T` агенты совершают заранее определенное количество действий.

Для ситуации, когда в системе было около 1000 агентов и 2370 потенциальных сценариев, экспериментально было подобрано число циклов, равное 15. При этом в системе остается довольно большое число потенциальных сценариев, с которыми никто из агентов учителей не взаимодействовал.

Фрагмент интерфейса, расположенный в модели справа от области, где происходит имитация создания и повторного использования цифрового контента, представлен на рисунке 52.

Граф связей, которые возникают между агентами, имитирующими поведение учителей, работающих над созданием и видоизменением контента в репозитории МЭШ, приведен на рисунке 53. Благодаря расширению NW среда NetLogo позволяет получить:

```
... - тут библиотека = подключаемое расширение  
(library) - library.
```

Однако среда многоагентного моделирования уступает в точности и воспроизводимости результатов средам, которые специально разработаны для статистического анализа данных. В данном проекте одной из задач было сопоставление результатов симуляции деятельности учителей в искусственном мире модели с результатами, основанными на реальных действиях учителей

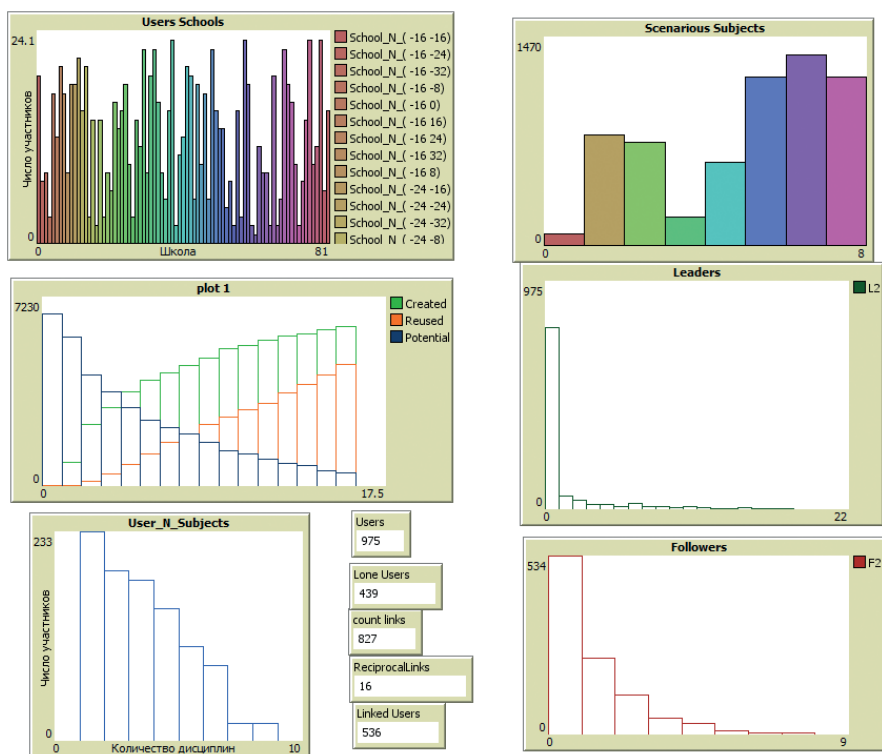


Рис. 52. Фрагмент интерфейса при визуализации распределений данных

Пояснения к графикам интерфейса

UserSchools — распределение учителей по школам.

ScenariosSubjects — распределение созданных участниками сценариев по предметам.

User_N_Subjects — количество учебных дисциплин, из которых отдельный участник может выбрать одну при взаимодействии с учебными сценариями.

Мы видим на гистограмме, что большинство участников связаны с одной дисциплиной; потом число дисциплин растет, а количество учителей, связанных с этими дисциплинами, падает. Система настроена таким образом, чтобы соответствовать реальному положению дел в репозитории, где большинство учителей связаны с одним или двумя предметами.

Plot1 — динамика создания и повторного использования учебных сценариев. Исходное число потенциально доступных для создания сценариев снижается, а число созданных и повторно использованных сценариев растет.

Leaders — гистограмма распределения участников-агентов по числу связей, которые к ним идут от других участников. На данном примере мы видим, что у большинства участников вообще нет никаких входящих связей. При этом есть один участник, который собрал 20 связей и является ключевым игроком, или лидером, в системе сознания и использования цифрового контента.

Followers — это гистограмма распределения участников-агентов по числу связей, которые идут от них к другим участникам. От 534-х — никаких связей. И один из авторов повторно использовал 8 объектов, созданных другими участниками.

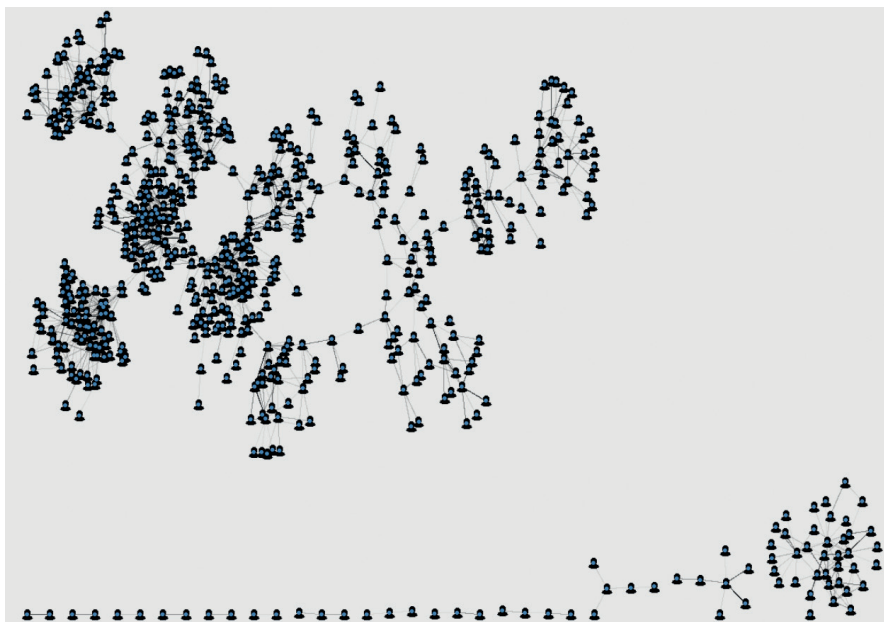


Рис. 53. Визуализация отношений между агентами, имитирующими поведение учителей

внутри цифрового репозитория МЭШ. Поскольку анализ и визуализация деятельности реального МЭШ-сообщества осуществляется в среде языка R, то и записи о действиях агентов внутри искусственного сообщества переносились и обрабатывались в среде языка R. К достоинствам модели следует отнести то, что она учитывала формирование бимодальной сети, в которой взаимодействия между участниками были опосредованы сценариями

уроков, которые они создают. Кроме того, в модели были учтены основные правила деятельности, которыми руководствуются участники в ходе работы на платформе МЭШ. Это позволило оперативно проводить эксперименты и получать в среде моделирования данные, которые оперативно переносились в среду R для анализа и визуализации. Например, мы могли проверить возможное влияние правила сегрегации учителей по количеству учебных дисциплин, которыми они могут отмечать сценарии уроков.

```
ifelse not Subj_Segregation
[ set wikilog lput (se [who] of player pagenum
"reuse" [School] of player Ball_Subj) wikilog ]
[ ifelse not member? Ball_Subj [User_Subj] of player
[ ] ;; если есть предпочтение по предмету, то с чужим
автором связь не образуется
[ set wikilog lput (se [who] of player pagenum
"reuse" [School] of player Ball_Subj) wikilog ] ]
```

В результате мы получили рабочую модель, пригодную для экспериментов над агентами, имитирующими сетевое сообщество создания и использования цифрового контента. Пример визуализации данных, полученных в искусственном сообществе, представлен на рисунке 53.

ЗАКЛЮЧЕНИЕ

Монография содержит множество карт и диаграмм, насыщена текстами программ на разных языках программирования, дополнена интерактивными моделями, которыми нужно управлять. Все эти расширения внутри печатного текста скорее огорчат, чем обрадуют читателя. Следовательно, в дополнение к тексту книги хочется сказать читателю, что изучать (тестировать) книгу лучше всего на вики-площадке, где собраны различные возможности для чтения.

Исходно мы рассматривали вики-площадку как сообщество практики, в котором представители естественно-научных дисциплин могли бы использовать статьи, данные, программы и диаграммы в качестве граничных объектов. Объединение и наращивание многообразия сущностей на поле вычислительной дидактики множеством участников предполагает создание разделяемой онтологии, которая была реализована в среде Semantic MediaWiki через создание категорий — классов объектов. Экспериментальная площадка использует технологию Semantic MediaWiki и главная особенность заключается в том, что мы думаем сразу о классах статей-объектов, которые относятся к определенной категории и обладают определенными свойствами. Вики-площадка или поле вычислительной дидактики открыто для совместного использования преподавателей и студентов по адресу: <http://digida.mgpi.ru> Интерактивный сбор и представление знаний при помощи форм, шаблонов и свойств системы совместной работы со знаниями Semantic MediaWiki, обеспечивает единообразный сбор данных о сущностях, представляющих новую область знаний и возможность поиска и формирования отчетов по запросам, сформулированным на языке ASK. Мы не просто создаем отдельные статьи, например, про понятие «сетевое сообщество» или «большие данные» или про языки обучения Scratch или Snap!, а исходно создаем классы, к которым относятся эти статьи. Концепции, модели и данные, о которых рассказывалось в книге, ждут читателей в вики.

ЛИТЕРАТУРА

Берроуз, Севидж 2016 — Берроуз Р., Севидж М. После кризиса? Big Data и методологические вызовы эмпирической социологии / пер. с англ. П. А. Максимова, Н. В. Романовского // Социологические исследования. 2016. № 3 (383). С. 28–35.

Воинов и др. 2019 — Воинов Н. В., К. Родригес Гарсон, Никифоров И. В., Дробинцев П. Д. Система обработки больших данных для анализа событий репозитория GitHub // Международная конференция по мягким вычислениям и измерениям. 2019. № 1. С. 283–286.

Дудко, Патаракин 2023 — Дудко В. В., Патаракин Е. Д. Исследование научных школ университета средствами библиометрического картирования // Территория новых возможностей. Вестник Владивостокского государственного университета. 2023. Т. 15. № 1. С. 150–167. <https://doi.org/10.24866/VVSU/2949-1258/2023-1/150-167>

Евин 2012 — Евин И. А. Сложные сети — новый инструмент изучения сложных систем // Сложные системы. 2012. № 2 (3). С. 66–74.

Ершов 1981 — Ершов А. П. Программирование — вторая грамотность. Новосибирск: ВЦ СО АН СССР, 1981. 18 с. (Препринт. / АН СССР, Сиб. отд-ние, ВЦ. 293).

Журавлева 2015 — Журавлева Е. Ю. Социология в сетевой среде: к цифровым социальным исследованиям // Социологические исследования. 2015. № 8 (376). С. 25–33.

Кнорр-Цетина 2002 — Кнорр-Цетина К. Объектная социальность: общественные отношения в постсоциальных обществах знания // Журнал социологии и социальной антропологии. 2002. Т. 5. № 1 (17). С. 101–124.

Кнорр-Цетина 2006 — Кнорр-Цетина К. Социальность и объекты. Социальные отношения в постсоциальных обществах знания // Социология вещей: сб. статей / под ред. В. Вахштайна. М.: Территория будущего, 2006. С. 267–306.

Латур 2013 — Латур Б. Наука в действии: следуя за учеными и инженерами внутри общества / [пер. с англ. К. Федоровой]. СПб.: Изд-во Европейского ун-та, 2013. 414 с. (Прагматический поворот; Вып. 6).

Латур 2014 — Латур Б. Пересборка социального: введение в акторно-сетевую теорию / пер. с англ. Ирины Полонской. М.: Изд. дом Высш. шк. экономики, 2014. 382 с. (Серия «Социальная теория»).

Латур 2015 — Латур Б. Пастер: Война и мир микробов с приложением «Несводимого» / [пер. с фр. А. В. Дьякова; Европейский ун-т в Санкт-Петербурге]. СПб.: Изд-во Европейского ун-та в Санкт-Петербурге, 2015. 315 с. (Прагматический поворот; вып. 7).

Ло 2006 — Ло Дж. Объекты и пространства / пер. с англ. В. Вахштайн // Социологическое обозрение. 2006. Т. 5. № 1. С. 30–42.

Макаров 2012 — Макаров В. Л. Искусственные общества // Экономика и математические методы. 2012. Т. 48. № 3. С. 3–20.

Макаров, Бахтизин 2009 — 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.

Макаров, Бахтизин 2015 — Макаров В. Л., Бахтизин А. Р. Современные методы прогнозирования последствий управленческих решений // Управленческое консультирование. 2015. № 7 (79). С. 12–24.

Мещерякова 2020 — Мещерякова Н. Н. Методология познания цифрового общества // Цифровая социология. 2020. Т. 3. № 2. С. 17–26. <https://doi.org/10.26425/2658-347X-2020-2-17-26>

Морено 1958 — Морено Дж. Л. Социометрия: Экспериментальный метод и наука об обществе: Подход к новой политической ориентации / пер. с англ. В. М. Корзинкина; ред. пер. и предисл. М. Ш. Бахитова. М.: Изд-во иностр. лит., 1958. 289 с. (Для научных библиотек).

Назарчук 2011 — Назарчук А. В. О сетевых исследованиях в социальных науках // Социологические исследования. 2011. № 1 (321). С. 39–51.

Олескин 2012 — Олескин А. В. Сетевые структуры в биосистемах и человеческом обществе. М.: URSS, сор. 2012. 301 с. (Синергетика: от прошлого к будущему; № 65).

Парандекар и др. 2019 — Парандекар С., Патаракин Е. Д., Яйла Г. Обучение детей программированию: залог развития человеческого капитала в XXI веке. Руководство для российских законодателей и практиков в области образования. М.: Алекс (ИП Поликанин А. А.), 2019. 164 с.

Патаракин и др. 2012 — Патаракин Е. Д., Ярмахов Б. Б., Буров В. В. Продвижение социальных инноваций через общественное конструирование документов // Образовательные технологии и общество. 2012. Т. 15. № 2. С. 517–535.

Патаракин, Буров 2022 — Патаракин Е. Д., Буров В. В. «Незримый колледж» МЭШ // Вестник МГПУ. Серия: Информатика и информатизация образования. 2022. № 2 (60). С. 38–52. <https://doi.org/10.25688/2072-9014.2022.60.2.04>

Abar et al. 2017 — Abar S., Theodoropoulos G. K., Lemarinier P., O'Hare G. M. P. Agent Based Modelling and Simulation tools: A review of the state-of-art software // Computer Science Review. 2017. Vol. 24. P. 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>

Barabasi 2003 — Barabasi A.-L. Linked: How Everything Is Connected to Everything Else and What It Means. Plume, 2003. 238 p.

Barabasi 2019 — Barabasi A. Twenty Years of Network Science: From Structure to Control // APS March Meeting Abstracts. 2019. P. S53-001.

Barnes 1954 — Barnes J. A. Class and Committees in a Norwegian Island Parish // Human Relations. 1954. Vol. 7. P. 39–58. <https://doi.org/10.1177/001872675400700102>

Baumer et al. 2017 — Baumer B. S., Kaplan D. T., Horton N. J. Modern Data Science with R. CRC Press, 2017. 582 p.

Blikstein 2013 — Blikstein P. Gears of Our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future // IDC'13: Proceedings of the 12th International Conference on Interaction Design and Children / Association for Computing Machinery. New York, NY, 2013. P. 173–182. <https://doi.org/10.1145/2485760.2485786>

Blikstein, Krannich 2013 — Blikstein P., Krannich D. The Makers' Movement and FabLabs in Education: Experiences, Technologies, and Research // IDC'13: Proceedings of the 12th International Conference on Interaction Design and Children / Association for Computing Machinery. New York, NY, 2013. P. 613–616. <https://doi.org/10.1145/2485760.2485884>

Börner et al. 2020 — Mapping the co-evolution of artificial intelligence, robotics, and the internet of things over 20 years (1998–2017) / K. Börner et al. // PLoS ONE. 2020. Vol. 15. e0242984. <https://doi.org/10.1371/journal.pone.0242984>

Börner et al. 2009 — Teaching Children the Structure of Science / K. Börner et al. // VDA 2009: Visualization and Data Analysis. San Jose, CA, 2009. P. 724307—724307. <https://doi.org/10.1117/12.812347>

Börner 2011 — Börner K. Plug-and-Play Macroscopes // Communications of the ACM. 2011. Vol. 54. Iss. 3. P. 60–69. <https://doi.org/10.1145/1897852.1897871>

Börner 2012 — Börner K. Visual Analytics in Support of Education // LAK'12: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge / Association for Computing Machinery. New York, NY, 2012. P. 2–3. <https://doi.org/10.1145/2330601.2330604>

Börner, Record 2017 — Börner K., Record E. Macroscopes for Making Sense of Science // PEARC17: Practice and Experience in Advanced Research Computing 2017 / Association for Computing Machinery. New York, 2017. P. 1–2. <https://doi.org/10.1145/3093338.3106387>

Brandes et al. 2016 — Brandes U., Holm E., Karrenbauer A. Cliques in Regular Graphs and the Core-Periphery Problem in Social Networks // Combinatorial Optimization and Applications / T.-H. H. Chan, M. Li, L. Wang (eds.). Cham: Springer, 2016. P. 175–186. (Lecture Notes in Computer Science. 10043). https://doi.org/10.1007/978-3-319-48749-6_13

Brandes 2020 — Brandes U. Central Positions in Social Networks // Computer Science — Theory and Applications, Lecture Notes in Computer Science / H. Fernau (ed.). Springer International Publishing, Cham, 2020. P. 30–45. https://doi.org/10.1007/978-3-030-50026-9_3

Brandes, Erlebach 2005 — Network Analysis: Methodological Foundations / edited by U. Brandes, T. Erlebach. Berlin: Springer Berlin Heidelberg, 2005. 484 p. (Lecture Notes in Computer Science. 3418).

Brennan, Resnick 2013 — Brennan, K., Resnick, M. Stories from the Scratch Community: Connecting with Ideas, Interests, and People // SIGCSE '13: Proceeding of the 44th ACM Technical Symposium on Computer Science Education / Association for Computing Machinery. New York, NY, 2013. P. 463–464. <https://doi.org/10.1145/2445196.2445336>

Cai et al. 2021 — Cai T., Chen C., Huang T.-H., Ritter F. E. What Makes A Good Reference Manager? A Quantitative Analysis of Bibliography Management Applications // Asian CHI Symposium 2021 / Association for Computing Machinery. New York, NY, 2021. P. 64–69. <https://doi.org/10.1145/3429360.3468183>

Cao et al. 2022 — Group Size and Group Performance in Small Collaborative Team Settings: An Agent-Based Simulation Model of Collaborative Decision-Making Dynamics. Complexity / S. Cao et al. // Complexity. 2022. 8265296. 16 pages. <https://doi.org/10.1155/2022/8265296>

Cao et al. 2023 — Visualizing Collective Idea Generation and Innovation Processes in Social Networks / Y. Cao et al. // IEEE Transactions on Computational Social Systems. 2023. Vol. 10. Iss. 5. P. 2234–2243. <https://doi.org/10.1109/TCSS.2022.3184628>

Choudhury et al. 2020 — GitLab: Work Where You Want, When You Want / P. Choudhury et al. // Journal of Organization Design. 2020. Vol. 9. Iss. 1. Article 23. <https://doi.org/10.1186/s41469-020-00087-8>

Cramer et al. 2015 — NetSci High: Bringing Network Science Research to High Schools / C. Cramer et al. // Complex Networks VI: Proceedings of the 6th Workshop on Complex Networks CompleNet 2015 / by ed. G. Mangioni, F. Simini, S. M. Uzzo, D. Wang. Springer, 2015. P. 209—218. (Studies in Computational Intelligence. Vol. 597).

Dasgupta et al. 2016 — Dasgupta S., Hale W., Monroy-Hernández A., Hill B. M. Remixing as a Pathway to Computational Thinking // CSCW'16: Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing / Association

for Computing Machinery. New York, NY, 2016. P. 1438–1449. <https://doi.org/10.1145/2818048.2819984>

De Caux et al. 2014 — Dynamic, small-world social network generation through local agent interactions / R. De Caux et al. // *Complexity*. 2014. Vol. 19. Iss. 6. P. 44—53. <https://doi.org/10.1002/cplx.21528>

Dechurch, Mesmer-Magnus 2010 — Dechurch L. A., Mesmer-Magnus J. R. Measuring Shared Team Mental Models: A Meta-Analysis // *Group Dynamics: Theory, Research, and Practice*. 2010. Vol. 14. Iss. 1. P. 1–14. <https://doi.org/10.1037/a0017455>

Donnellan 2020 — Donnellan L. Archaeological Networks and Social Interaction // *Archaeological Networks and Social Interaction* / L. Donnellan (ed.). London; New York: Routledge, 2020. P. 1–20. (Routledge Studies in Archaeology).

Eck, Waltman 2009 — Eck N. J. van, Waltman L. Software Survey: VOSviewer, a Computer Program for Bibliometric Mapping // *Scientometrics*. 2009. Vol. 84. Iss. 2. P. 523–538. <https://doi.org/10.1007/s11192-009-0146-3>

Engelbart 1988 — Engelbart D. C. A Conceptual Framework for the Augmentation of Man's Intellect // *Computer-Supported Cooperative Work: A Book of Readings* / I. Greif (ed.). San Francisco, CA: Morgan Kaufmann Publishers Inc., 1988. P. 35–65.

Engeström 2005 — Engeström J. Why Some Social Network Services Work and Others Don't — Or: The Case for Object-Centered Sociality (2005) // Zengestrom: site. URL: <http://www.zengestrom.com/blog/2005/04/why-some-social-network-services-work-and-others-dont-or-the-case-for-object-centered-sociality.html> (дата обращения: 04.07.2024).

Epstein 2006 — Epstein J. M. Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press, 2006. 352 p. <https://www.jstor.org/stable/j.ctt7rxj1>

Epstein 2013 — Epstein J. M. *Agent_Zero: Toward Neurocognitive Foundations for Generative Social Science*. Princeton University Press, 2013. 192 p. <https://doi.org/10.2307/j.ctt5hnp5x>

Epstein, Axtell 1996 — Epstein J. M., Axtell R. Growing Artificial Societies: Social Science from the Bottom Up / The Brookings Institution. Washington, DC, 1996. 228 p.

Fernandez 2011 — Fernandez P. Zotero: Information Management Software 2.0 // Library Hi Tech News. 2011. Vol. 28. Iss. 4. P. 5–7. <https://doi.org/10.1108/07419051111154758>

Fontana, Terna 2015 — Fontana M., Terna P. From Agent-Based Models to Network Analysis (and Return): The Policy-Making Perspective // Journal on Policy and Complex Systems. 2014. Vol. 2. <https://doi.org/10.18278/jpcs.2.1.8>

Franklin 2015 — Franklin D. Putting the Computer Science in Computing Education Research // Communications of the ACM. Vol. 58. Iss. 2. P. 34–36. <https://doi.org/10.1145/2700376>

Gabora, Saberi 2011 — Gabora L., Saberi M. An Agent-Based Model of the Cognitive Mechanisms Underlying the Origins of Creative Cultural Evolution // C'11: Proceedings of the 8th ACM Conference on Creativity and Cognition, C&A / Association for Computing Machinery. New York, NY, 2011. P. 299–306. <https://doi.org/10.1145/2069618.2069667>

Gama et al. 2014 — Network Analysis and Intra-Team Activity in Attacking Phases of Professional Football / J. Gama et al. // International Journal of Performance Analysis in Sport. 2014. Vol. 14. P. 692–708. <https://doi.org/10.1080/24748668.2014.11868752>

Garcia et al. 2009 — Rediscovering the Passion, Beauty, Joy, and Awe: Making Computing Fun Again, Continued / D. D. Garcia et al. // ACM SIGCSE Bulletin. Vol. 40. Iss. 1. P. 217–218. <https://doi.org/10.1145/1539024.1508889>

Garcia et al. 2015 — Garcia D., Harvey B., Barnes T. The Beauty and Joy of Computing // ACM Inroads. 2015. Vol. 6. Iss. 4. P. 71–79. <https://doi.org/10.1145/2835184>

García-Peña et al. 2019 — García-Peña C., Gutiérrez-Robledo L. M., Cabrera-Becerril A., Fajardo-Ortiz D. Team Assembly Mechanisms and the Knowledge Produced in the Mexico's National Institute of Geriatrics: A Network Analysis and Agent-Based Modeling Approach // Scientifica. 2019. e9127657. <https://doi.org/10.1155/2019/9127657>

Gilbert 2008 — Gilbert N. Agent-Based Models. Sage Publications, Inc., 2008. 98 p. (Series “Quantitative Applications in the Social Sciences”. 153).

Gómez-Zarà et al., 2020 — Gómez-Zarà D., Dechurch L. A., Contractor N. S. A Taxonomy of Team-Assembly Systems: Understanding How People Use Technologies to Form Teams // Proceedings of the ACM on Human-Computer Interaction. 2020. Vol. 4. P. 1–36. <https://doi.org/10.1145/3415252>

Guimerà et al. 2005 — Guimerà R., Uzzi B., Spiro J., Amaral L. A. N. Team Assembly Mechanisms Determine Collaboration Network Structure and Team Performance // Science. Vol. 308. P. 697–702. <https://doi.org/10.1126/science.1106340>

Harel, Papert 1991 — Constructionism: Research Reports and Essays, 1985–1990 / edited by I. Harel, S. Papert. Praeger, 518 p.

Hohenwarter, Lavicza 2011 — Hohenwarter M., Lavicza Z. The Strength of the Community: How GeoGebra can Inspire Technology Integration in Mathematics // Model-Centered Learning / edited by L. Bu, R. Schoen. SensePublishers, 2011. P. 7–12. (Modeling and Simulations for Learning and Instruction, vol. 6). https://doi.org/10.1007/978-94-6091-618-2_2

Huerta-Quintanilla 2013 — Huerta-Quintanilla R., Canto-Lugo E., Viga-de Alva D. Modeling Social Network Topologies in Elementary Schools // PloS One. 2013. Vol. 8. Iss. 2. e55371. <https://doi.org/10.1371/journal.pone0055371>

Kafai 2016 — Kafai Y. From Computational Thinking to Computational Participation in K – 12 Education // Communications of the ACM. 2016. Vol. 59. Iss. 8. P. 26–27. <https://doi.org/10.1145/2955114>

Kallinikos et al. 2013 — Kallinikos J., Aaltonen A., Marton A. The Ambivalent Ontology of Digital Artifacts // MIS Quarterly. 2013. Vol. 37. Iss. 2. P. 357–370. <https://doi.org/10.25300/MISQ/2013/37.2.02>

Kallinikos J., Mariátegui 2011 — Kallinikos J., Mariátegui J.-C. Video as Digital Object: Production and Distribution of Video Content in the Internet Media Ecosystem // The Information Society. 2011. Vol. 27. Iss. 5. P. 281–294. <https://doi.org/10.1080/01972243.2011.607025>

Kaschesky, Selmi 2013 — Kaschesky M., Selmi L. Fusepool R5 Linked Data Framework: Concepts, Methodologies, and Tools for Linked Data // Dg.o’13: Proceedings of the 14th Annual International Conference on Digital Government Research / Association

for Computing Machinery, New York, NY, 2013. P. 156–165. [https://doi.org/ 10.1145/2479724.2479748](https://doi.org/10.1145/2479724.2479748)

Kröckel et al. 2017 — Kröckel P., Piazza A., Neuhofer K. Dynamic Network Analysis of the Euro2016 Final: Preliminary Results // FiCloudW: 5th International Conference on Future Internet of Things and Cloud Workshops. Prague, 2017. P. 114–119. <https://doi.org/10.1109/FiCloudW.2017.98>

Law 1984 — Law J. On the Methods of Long-Distance Control: Vessels, Navigation and the Portuguese Route to India // The Sociological Review. 1984. Vol. 32. P. 234–263. DOI: 10.1111/j.1467-954X.1984.tb00114.x

Leuf 2006 — Leuf B. The Semantic Web: Crafting Infrastructure for Agency. 1st ed. Wiley & Sons Ltd., 2006. 376 p.

Lorig, Timm 2020 — Lorig F., Timm I. J. Simulation-Based Data Acquisition // Principles of Data Science, Transactions on Computational Science and Computational Intelligence / edited by H. R. Arabnia et al. Cham: Springer, 2020. P. 1–15. https://doi.org/10.1007/978-3-030-43981-1_1

Luke 2015 — Luke D. A User's Guide to Network Analysis in R. Springer, 2015. 238 p.

Margolis, Parboosingh 2015 — Margolis A., Parboosingh J. Networked Learning and Network Science: Potential Applications to Health Professionals' Continuing Education and Development // The Journal of Continuing Education in the Health Professions. 2015. Vol. 35. № 3. P. 211–219. <https://doi.org/10.1002/chp.21295>

Monroy-Hernández 2013 — Monroy-Hernández A. Designing for Remixing: Supporting an Online Community of Amateur Creators: PhD thesis. Massachusetts, 2013. 176 p. URL: <https://dspace.mit.edu/handle/1721.1/78202> (дата обращения: 09.07.2024).

Moreno 1934 — Moreno J. L. Who Shall Survive?: A New Approach to the Problem of Human Interrelations. Nervous and Mental Disease Publishing Company, 1934. 440 p.

Oliveira et al. 2018 — Oliveira N., Muller M., Andrade N., Reinecke K. The Exchange in StackExchange: Divergences Between Stack Overflow and Its Culturally Diverse Participants // Proceedings

of the ACM on Human-Computer Interaction. Vol. 2 (CSCW). Article 130. 22 pages. <https://doi.org/10.1145/3274399>

Patarakin et al. 2019 — Patarakin E., Burov V., Yarmakhov B. Computational Pedagogy: Thinking, Participation, Reflection // Digital Turn in Schools — Research, Policy, Practice, Lecture Notes in Educational Technology / edited by T. Väljataga, M. Laanpere. Singapore: Springer, 2019. P. 123—137. https://doi.org/10.1007/978-981-13-7361-9_9

Patarakin et al. 2021 — Patarakin E., Vachkova S., Burov V. Agent-Based Modeling of Teacher Interaction Within a Repository of Digital Objects // SHS Web Conference. Vol. 98. e05013. <https://doi.org/10.1051/shsconf/20219805013>

Patarakin 2017 — Patarakin E. D. Wikigrams-Based Social Inquiry // Digital Tools and Solutions for Inquiry-Based STEM Learning / edited by I. Levin and D. Tsybulsky. IGI Global, 2017. P. 112–138. <https://doi.org/10.4018/978-1-5225-2525-7.ch005>

Patarakin, Shilova 2015 — Patarakin Ye., Shilova O. Concept of Learning Design for Collaborative Network Activity // Procedia — Social and Behavioral Sciences. Vol. 214. P. 1083–1090. <https://doi.org/10.1016/j.sbspro.2015.11.709>

Payette 2020 — Payette N. Collaborating Like Professionals: Integrating NetLogo and GitHub // Advances in Social Simulation, Springer Proceedings in Complexity / edited by H. Verhagen, M. Borit, G. Bravo, N. Wijermans. Cham: Springer, 2020. P. 343–348. https://doi.org/10.1007/978-3-030-34127-5_33

Peeples 2019 — Peeples M. A. Finding a Place for Networks in Archaeology // Journal of Archaeological Research. 2019. Vol. 27. P. 451–499. <https://doi.org/10.1007/s10814-019-09127-8>

Railsback, Grimm 2019 — Railsback S. F., Grimm V. Agent-Based and Individual-Based Modeling: A Practical Introduction. Second Edition. Princeton University Press, 2019. 360 p.

Rakić et al. 2020 — Rakić, K., Rosić, M., Boljat, I. A Survey of Agent-Based Modelling and Simulation Tools for Educational Purpose // Tehnički vjesnik. Vol. 27. Iss. 3. P. 1014–1020. <https://doi.org/10.17559/TV-20190517110455>

Reeves, Nass 2003 — Reeves B., Nass C. The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places. Cambridge University Press, 2003. 317 p.

Repenning 2013 — Repenning A. Conversational Programming: Exploring Interactive Program Analysis // Proceedings of the 2013 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software, Onward! 2013 / Association for Computing Machinery, New York, NY, 2013. P. 63–74. <https://doi.org/10.1145/2509578.2509591>

Repenning, Basawapatna 2016 — Repenning A., Basawapatna A. Drops and Kinks: Modeling the Retention of Flow for Hour of Code Style Tutorials // WiPSCE'16: Proceedings of the 11th Workshop in Primary and Secondary Computing Education / Association for Computing Machinery. New York, NY, 2016. P. 76–79. <https://doi.org/10.1145/2978249.2978260>

Resnick 1997 — Resnick M. Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. MIT Press, 1997. 184 p.

Resnick 2017 — Resnick M. Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play. MIT Press, 2017. 208 p.

Rodriguez et al. 2018 — Courseware as Code Setting a new bar for transparency and collaboration / J. Rodriguez et al. // IEEE Frontiers in Education Conference 2018 (FIE). San Jose, CA: IEEE, 2018. P. 1–4. <https://doi.org/10.1109/FIE.2018.8658928>

Rodriguez et al. 2020 — Rodriguez J. M., Allison B. J., Apsey C. W., Boudreau T. M. Courseware as Code: Instituting Agile Courseware Collaboration // IEEE Security & Privacy. 2020. Vol. 18. Iss. 6. P. 59–62. <https://doi.org/10.1109/MSEC.2020.3004960>

Rollins et al. 2014 — A Computational Model Library for Publishing Model Documentation and Code / N. D. Rollins et al. // Environmental Modelling & Software. 2014. Vol. 61. P. 59–64. <https://doi.org/10.1016/j.envsoft.2014.06.022>

Rosnay 1979 — Rosnay J. de. The macroscope: A new world scientific system, 1st ed. New York: Harper & Row, 1979. 247 p.

Ruf et al. 2014 — Ruf A., Mühling A., Hubwieser P. Scratch vs. Karel: impact on learning outcomes and motivation // Proceedings of the 9th Workshop in Primary and Secondary Computing Education / Association for Computing Machinery. Berlin, 2014. P. 50–59. <https://doi.org/10.1145/2670757.2670772>

Safari et al. 2020 — Safari H., Sabri N., Shahsavan F., Bahrak B. An Analysis of GitLab’s Users and Projects Networks // 10th International Symposium on Telecommunications (IST). Tehran, 2020. P. 194–200. <https://doi.org/10.1109/IST50524.2020.9345844>

Sayama et al. 2016 — What are Essential Concepts About Networks? / H. Sayama et al. // Journal of Complex Networks. 2016. Vol. 4. Iss. 3. P. 457–474. <https://doi.org/10.1093/comnet/cnv028>

Sayama H. 2015 — Sayama H. Introduction to the Modeling and Analysis of Complex Systems. Geneseo, NY: Open SUNY, 2015. 498 p.

Sedita et al. 2020 — Sedita S. R., Caloffi A., Lazzeretti L. The Invisible College of Cluster Research: A Bibliometric Core-Periphery Analysis of the Literature // Industry and Innovation. 2020. Vol. 27. Iss. 2. P. 562–584. <https://doi.org/10.1080/13662716.2018.1538872>

Serres 2007 — Serres M. The Parasite / translated by L. R. Schehr. University of Minnesota Press, 2007. 255 p.

Sheetz et al. 2015 — Sheetz L., Dunham V., Cooper J. Professional Development for Network Science as a Multi-Disciplinary Curriculum Tool // IEEE Integrated STEM Education Conference 2015 (ISEC). Princeton, NJ, 2015. P. 178–182. <https://doi.org/10.1109/ISECon.2015.7119919>

Sporns 2011 — Sporns O. The Human Connectome: A Complex Network // Annals of the New York Academy of Sciences. 2011. Vol. 1224. P. 109–125. <https://doi.org/10.1111/j.1749-6632.2010.05888.x>

Vendome et al. 2020 — Vendome C., Rao D. M., Giabbanelli P. J. How Do Modelers Code Artificial Societies? Investigating Practices and Quality of Netlogo Codes from Large Repositories // 2020 Spring Simulation Conference (SpringSim). Fairfax, VA, 2020. P. 1–12. <https://doi.org/10.22360/SpringSim.2020.HSAA.007>

Wang et al. 2014 — An Analytical Framework for Understanding Knowledge-Sharing Processes in Online Q&A Communities / G. A. Wang et al. // ACM Transactions on Management Information Systems (TMIS). Vol. 5. Iss. 4. Article 18. P. 1–31. <https://doi.org/10.1145/2629445>

Wang 2008 — Wang S. Ontology of Learning Objects Repository for Pedagogical Knowledge Sharing // Interdisciplinary Journal of E-Learning and Learning Objects. 2008. Vol. 4. № 1. P. 1–12.

Wäsche et al. 2017 — Wäsche H., Dickson G., Woll A., Brandes U. Social Network Analysis in Sport Research: An Emerging Paradigm // European Journal for Sport and Society. 2017. Vol. 14. № 3. P. 138–165. <https://doi.org/10.1080/16138171.2017.1318198>

Watts 2003 — Watts D. J. Small Worlds: The Dynamics of Networks between Order and Randomness / illustrated edition. Princeton University Press, 2003. 262 p.

Watts 2004 — Watts D. J. Six Degrees: The Science of a Connected Age. W. W. Norton & Company, 2004. 384 p.

Weintrop, Wilensky 2015 — Weintrop D., Wilensky U. Using Commutative Assessments to Compare Conceptual Understanding in Blocks-Based and Text-Based Programs // ICER'15: Proceedings of the Eleventh Annual International Conference on International Computing Education Research / Association for Computing Machinery. New York, NY, 2015. P. 101–110. <https://doi.org/10.1145/2787622.2787721>

Wenger et al. 2002 — Wenger E., McDermott R. A., Snyder W. Cultivating Communities of Practice: A Guide to Managing Knowledge. Harvard Business Press, 2002. 284 p.

Wenger 1999 — Wenger E. Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, 1999. 318 p.

Whitney 2012 — Whitney H. Data Insights: New Ways to Visualize and Make Sense of Data. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2012. 320 p.

Wickham 2015 — Wickham H. Advanced R. CRC Press, 2015. 476 p.

Wickham, Grolemund 2016 — Wickham H., Grolemund G. R. for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media, Inc., 2016. 492 p.

Wild 2016 — Wild F. Learning Analytics in R with SNA, LSA, and MPIA. Springer, 2016. 275 p.

Wilensky, Rand 2015 — Wilensky U., Rand W. An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo. MIT Press, 2015. 504 p.

Will 2016 — Will T. A Flocking Model of Human Organizing: Leadership Implications // Academy of Management Proceedings. 2016. Iss. 1. P. 13091. <https://doi.org/10.5465/ambpp.2016.13091abstract>

Will 2016a — Will T. E. Flock Leadership: Understanding and Influencing Emergent Collective Behavior // The Leadership Quarterly. Vol. 27. P. 261–279. <https://doi.org/10.1016/j.leaqua.2016.01.002>

Williams 2020 — Williams B. Dimensions & VOSviewer Bibliometrics in the Reference Interview // The Code4Lib Journal. 2020. Vol. 47. e14964. URL: <https://journal.code4lib.org/articles/14964>

Youngman, Hadzikadic 2014 — Complexity and the Human Experience: Modeling Complexity in the Humanities and Social Sciences / edited by P. A. Youngman, M. Hadzikadic. New York: Jenny Stanford Publishing, 2014. 304 p. <https://doi.org/10.1201/b16877>

Е. Д. Патаракин

**ИЗУЧЕНИЕ И МОДЕЛИРОВАНИЕ
СОВМЕСТНОЙ ДЕЯТЕЛЬНОСТИ
В СЕТЕВЫХ СООБЩЕСТВАХ**

Монография

Редактор: *И. Е. Посоха*

Технический редактор: *О. Г. Арефьева*

Верстка: *А. В. Бармин*

Формат: 60 × 90 1/16. Объем: 8,75 усл. печ. л.

Тираж: 500 экз.

Московский городской педагогический университет
Научно-информационный издательский центр
129226, Москва, 2-й Сельскохозяйственный пр., 4