

**МИГРАЦИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В ОБЛАКО****А.В. Борщев (Санкт-Петербург)****О текущем состоянии имитационного моделирования и тенденциях**

Как мы уже говорили два года назад [1], мир “имитационного моделирования (ИМ) для бизнеса” небольшой и расширяется медленно. Основное препятствие для роста популярности имитационных моделей – сложность их создания. Вторая проблема – отсутствие устойчивого спроса. У бизнеса нет однозначного понимания, какой именно класс проблем решается при помощи ИМ, всегда найдётся кто-то, кто скажет, что прикинуть можно и в Excel, а оптимизировать – линейным решателем. В целом, мы продолжаем наблюдать небыстрый рост спроса при небольшом, но стабильном сообществе разработчиков имитационных моделей.

В области методов и языков имитационного моделирования также происходит не много интересного. Последним значительным событием было появление и начало практического использования агентного моделирования (как самого по себе, так и в комбинации с традиционными подходами) [2]. С тех пор прошло больше десяти лет, но ничего сопоставимого по масштабу пока не случилось. На рынке инструментов ИМ остаются те же игроки, постепенно улучшающие (или не улучшающие) свои продукты.

Единственная интересная тенденция последних лет – миграция ИМ в облако. Это и есть наша тема сегодня. Появились относительно дешёвые, доступные, надёжные, безопасные и легко масштабируемые ресурсы, расположенные в интернете (например, [8]). Продажа облачных сервисов стандартизирована, ценообразование прозрачно. На стороне пользователя точно так же стал стандартом новый тип интерфейса – браузерный, и появились мощные средства его разработки, такие как HTML5. В этих обстоятельствах производители десктопного программного обеспечения (ПО), естественно, стали пробовать полностью или частично переносить приложения в облако. Это затронуло даже такой инерционный сегмент, как инструменты ИМ общего назначения для не инженерных приложений (в инженерном моделировании облако уже стало стандартом [9]).

**Что, собственно, уходит в облако?**

Давайте посмотрим на элементы жизненного цикла имитационной модели. Это разработка и отладка, перемешанная с работой с данными, верификацией и валидацией, дизайн экспериментов, дизайн интерфейса конечного пользователя (dashboards), собственно выполнение имитационных экспериментов, анализ и экспорт результатов, возвращение на любую из предыдущих фаз и т.д. На каких-то фазах участвует только разработчик (консультант), на каких-то – только заказчик, где-то они работают совместно.

Разработка имитационных моделей, включая их отладку, – это интенсивное прецизионное графическое редактирование, ввод текста, работа с иерархически организованными проектами, одновременное присутствие нескольких окон и большого количества информации на экране. Это требует наличия экрана большой площади, клавиатуры и мыши, так что планшеты и телефоны для этой задачи отпадают. Остаются ноутбуки и десктопы – а там, в общем, несложно установить и традиционное офлайновое приложение. Его минусы (такие как необходимость установки на новом устройстве и, иногда, явных действий пользователя при апгрейде) в этом контексте не очень существенны.

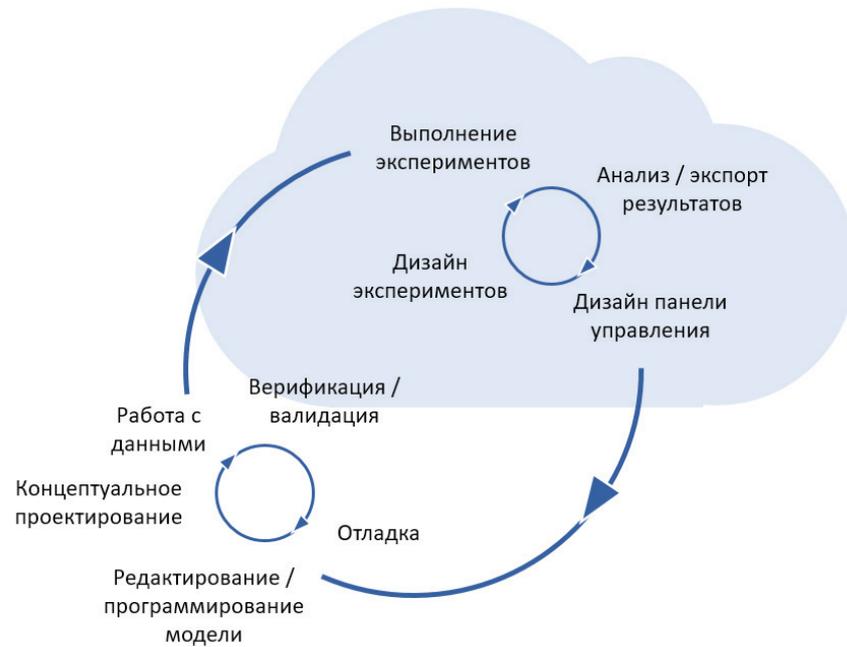


Рис. 1. Что из жизненного цикла имитационной модели уйдёт в облако

Примеры похожих приложений, так и не перешедших в браузер: Adobe Photoshop, Adobe Illustrator, почти весь Microsoft Office, инструменты разработки ПО, например, JetBrains Idea и т.п. Кроме того, у офлайна есть и плюсы: проще вызывать модули, написанные на “не браузерных” языках (нативный код) и получать доступ к данным.

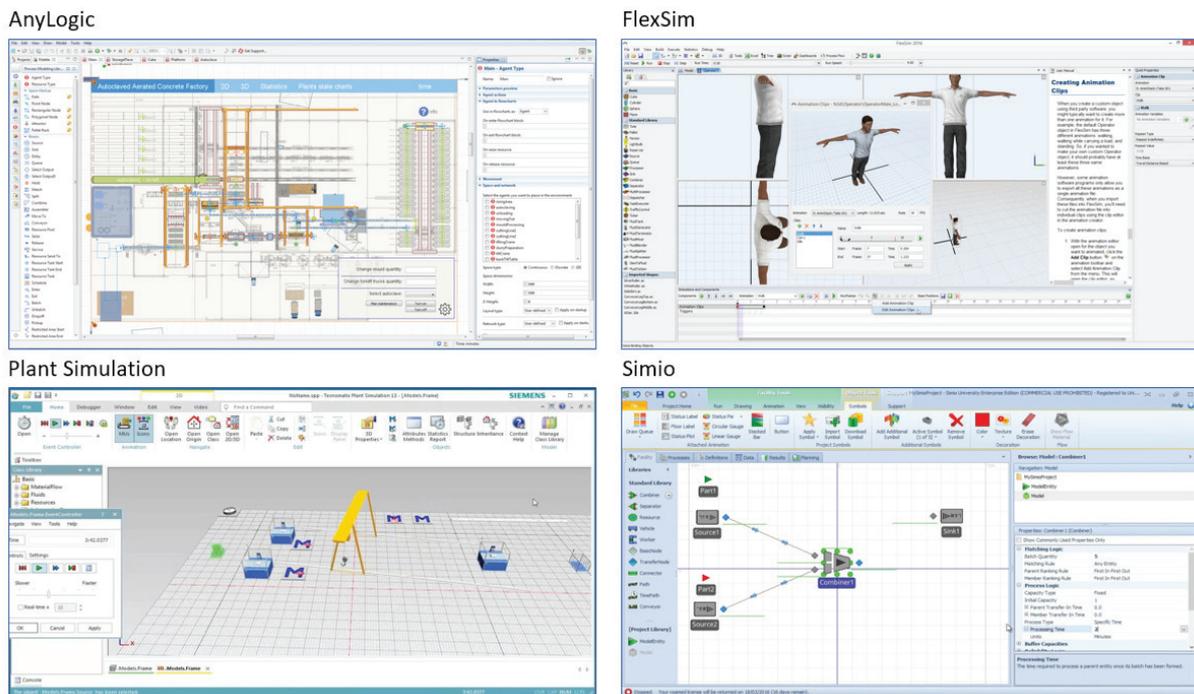


Рис. 2. Интерфейсам такого типа сейчас нет большого смысла переходить в браузер

Судя по всему, средства разработки имитационных моделей (редакторы и отладчики, так называемые authoring tools), за исключением очень упрощённых типа Insight Maker [3], в ближайшем будущем останутся обычными десктопными приложениями и в браузер не перейдут. Всё остальное – уже в облаке или стремится туда.

## Облако как инструмент командной разработки, хранения и контроля версий

Здесь разработка ИМ не имеет существенных особенностей по сравнению с обычной разработкой ПО. Для хранения и контроля версий давно используются облачные средства типа GitHub [4]. Для облегчения командной разработки моделей желательно, чтобы проект делился на отдельные файлы, чтобы эти файлы были текстовые, например, формата XML и существовали качественные средства их сравнения и слияния.

Нельзя сказать, чтобы вендоры инструментов ИМ обращали много внимания на поддержку коллективной разработки. Плюс, впрочем, является то, что на сегодняшний день большинство инструментов ИМ либо используют XML как нативный формат файла модели (например, AnyLogic, Simio), либо имеют опцию сохранения в XML (FlexSim, Simul8); бинарники становятся архаизмом. Некоторые продукты имеют встроенную поддержку “необлачных” систем контроля версий, таких, как SVN. Встроенной в продукт поддержки Git никто не предоставляет, так что если хочется иметь имитационный проект в Git, нужно использовать отдельный Git-клиент. AnyLogic, кроме того, предоставляет простую систему хранения версий в AnyLogic Cloud [5].

## Облако как вычислительные мощности

Имитационные модели – вычислительно затратные объекты, и естественно использовать преимущества облака для повышения эффективности работы с ними.

В среднем, предоставляемый стандартным облачным сервисом узел уже более мощный, чем то, что стоит у вас на столе (от 8 ядер, от 16 Гб памяти). Во-вторых, его просто конфигурировать и менять. В-третьих, и это главное, этот узел легко масштабируется под ваши текущие потребности. Так что, если вам вдруг понадобится 20 или 100 узлов, работающих параллельно, они могут быть быстро выделены, а затем “отпущены”.

В связи с этим полезно разобраться с тем, как параллельные/распределённые вычисления используются в имитационном моделировании. Важно понимать, что так называемое “распределённое моделирование” (distributed simulation), когда выполнение одного прогона модели распараллеливается на несколько процессоров, которые поддерживают синхронизацию по модельному времени, – всегда экзотика и точечные решения, и ни в коем случае не стандарт в нашей области, то есть в ИМ для бизнеса. В общем случае, затраты на синхронизацию и обмен данными сводят на нет выигрыш от параллельной обработки. Понять, как именно можно разделить модель на “слабо связанные” части и разнести их, на настоящий момент только инженер высокой квалификации, и поэтому инструменты ИМ используют такой тип параллельности по минимуму. Например, AnyLogic использует multithreading (то есть задействует несколько доступных ядер процессора) при моделировании пешеходных потоков.

Зато использование множества параллельно работающих узлов естественно, просто реализуемо и даёт огромный выигрыш по времени для “многoproгонных” экспериментов, то есть для:

- Монте-Карло первого порядка – исследования влияния внутренней стохастичности модели на её выход;
- Монте-Карло второго порядка – исследования влияния нестабильных входных параметров на выход, или исследования поведения системы на огромном пространстве параметров, где регулярный перебор невозможен или нерационален;
- Варьирования параметров – получения зависимости выхода от одного или нескольких входов при регулярном их варьировании;
- Оптимизации – нахождения комбинации входных параметров, при которой достигается хорошее значение целевой функции при соблюдении ограничений.

При этом наибольший выигрыш от распараллеливания получается для экспериментов с абсолютно независимыми прогонами – то есть для всех, кроме оптимизации. Оптимизация требует знания результатов предыдущих прогонов для запуска последующих, поэтому какие-то прогоны по определению будут выполнены последовательно.

Архитектурно в облачном сервисе по выполнению таких экспериментов с моделями всегда присутствует диспетчер по распределению прогонов по узлам и балансировщик нагрузки (load balancer).

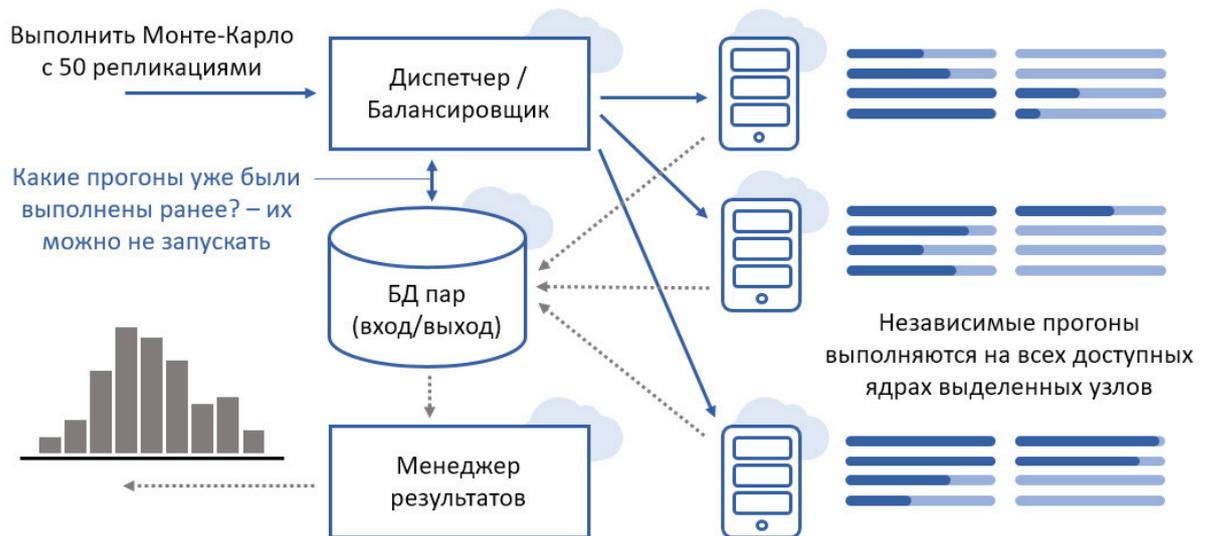


Рис. 3. Параллельное выполнение многопрогонных экспериментов

Стоит также напомнить, что запуская эксперимент в облаке, вы можете спокойно выключить компьютер и уйти, а затем в любое время проверить статус и просмотреть результаты.

### **Браузер как универсальное средство работы с моделью**

Если использование браузера как интерфейса при разработке модели возможно, но не даёт больших преимуществ, то работу с уже созданной моделью браузер выводит на новый уровень. Типичный сценарий такой работы включает:

- работу с версиями модели;
- задание/изменение входных параметров;
- наблюдение анимированного прогона и интерактивное взаимодействие с ним;
- планирование сложных экспериментов;
- конфигурирование панели управления (dashboard) с входами и выходами модели, которая скроет от конечного пользователя техническую начинку;
- выполнение экспериментов;
- просмотр, анализ (например, сравнение) и экспорт результатов.

Все эти действия отлично ложатся на браузерный frontend и при надлежащей вёрстке будут одинаково удобны везде – от телефона с тачскрином до десктопа с мышью. Это означает, что после того как модель загружена в облако, она становится мгновенно доступна в любой точке (где есть интернет) и с любого устройства (где есть браузер).

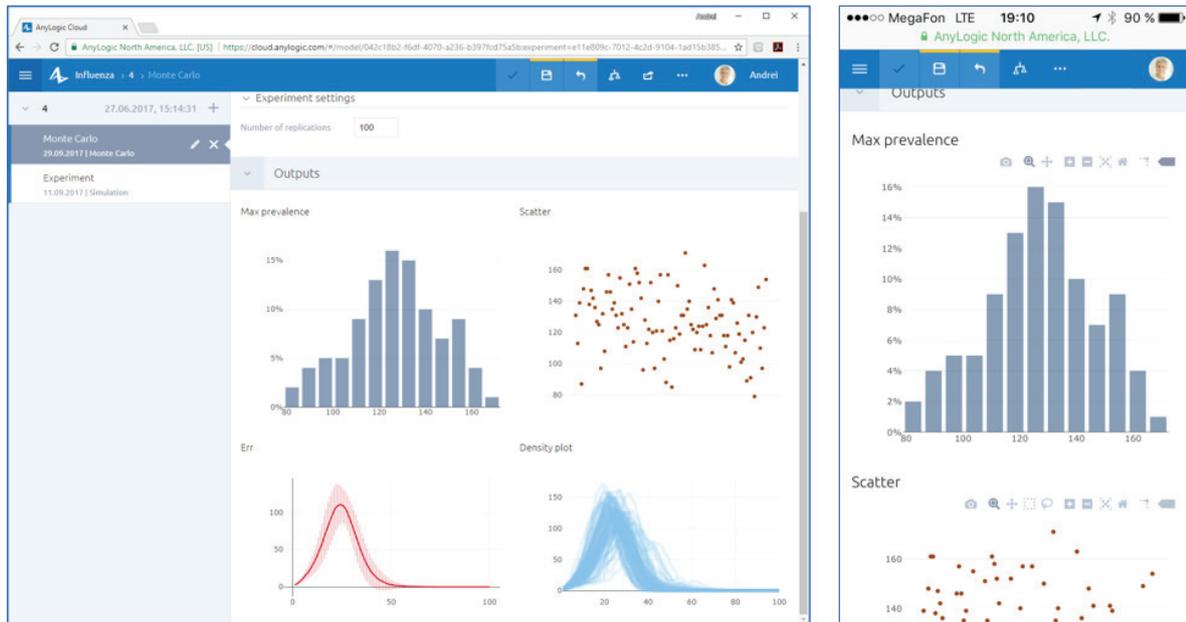


Рис. 4. Панель управления ИМ на экране ноутбука (Chrome) и iPhone (Safari)

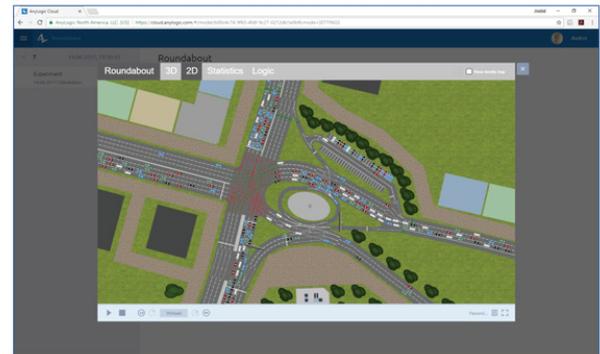
Отдельно хочу сказать о доставке анимации удалённо выполняющейся модели и реализации её интерактивности. Сцена имитационной модели, двумерная и/или трёхмерная, может содержать десятки тысяч элементов: представьте, например, более или менее детальную модель производства, склада или модель пешеходных потоков. Эта сцена непредсказуемо меняется в произвольные моменты времени. Изменения, естественно, появляются в облаке на сервере, где модель выполняется. И мы хотим их видеть на удалённом клиенте в браузере с минимальной задержкой, чтобы иметь возможность интерактивно управлять моделью во время её выполнения.

Съёмка и передача видео в данном случае не вариант, так как предполагает буферизацию (то есть, задержку) и снижает качество. Остаётся передача векторного изображения и его рендеринг в браузере. К нашей большой радости, в HTML5 появились две графические технологии, которые можно использовать для анимации ИМ: canvas (растровый холст) и SVG (Scalable Vector Graphics). AnyLogic Cloud использует canvas для 3D-анимации и SVG для 2D-анимации. SVG имеет массу преимуществ: легко масштабируемая картинка без потери качества, возможность привязки отклика на действия пользователя к любым элементам и т.д. Для оптимизации передачи изображения передаются только изменения векторной сцены, что позволяет существенно снизить трафик, повысить скорость отрисовки и сделать время отклика почти таким же, как при традиционном выполнении модели и анимации на одной машине. Simul8 (вторая компания, реализовавшая удалённую анимацию) использует только canvas и, видимо, похожую схему передачи изображения [6].

AnyLogic Cloud – Aerated Concrete Plant (3D view)



AnyLogic Cloud – An Intersection (2D view)



Simul8 Studio – Emergency Department (2.5D view)



Simul8 Studio – Job Shop (2.5D view)

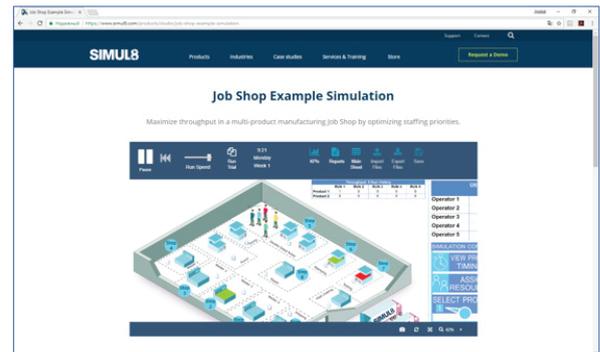


Рис. 5. Анимация ИМ в браузере в AnyLogic Cloud (canvas и SVG) и Simul8 (canvas)

### Облако как способ доставки моделей заказчику

Думать об облаке как об удалённых вычислительных мощностях, куда можно отправить что-то “посчитать”, а потом получить результат обратно, – ограниченный взгляд на вещи: облако в принципе меняет модель работы пользователя с приложениями. В случае ИМ меняется схема взаимодействия с заказчиком, а также любая коллективная работа с моделями.

Традиционный сценарий передачи заказчику результатов имитационного проекта (или итерации промежуточных фаз) предполагал встречу в офлайне, демонстрацию с компьютера разработчика, иногда установку у заказчика того же приложения для разработки или для просмотра ИМ, копирование массы файлов. Часто при этом могла потребоваться помощь системного администратора. Или же заказчику передавался просто отчёт.

Всё это уходит в прошлое. Если модель вместе с данными загружена в облако, достаточно просто выслать заказчику ссылку, по которой он сам сможет запустить заранее сконфигурированные эксперименты, создать новые, поменять параметры, посмотреть анимацию. Доступ к новой версии модели занимает секунды, всё выполняется в браузере, не нужно устанавливать никакого софта и назначать встречу. Запуск и демонстрация модели, просмотр и анализ результатов становятся возможным в любом месте, например, в кафе за завтраком при помощи телефона.

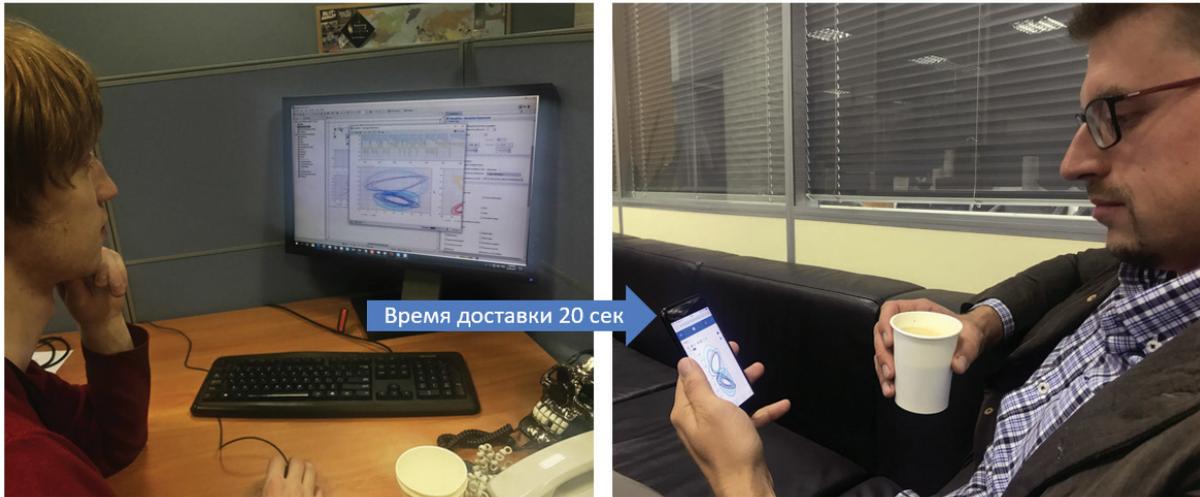


Рис. 6. Имитационная модель доставлена клиенту

### Облако как платформа для коллективной работы с моделями

Загруженная в облако с небольшой инфраструктурой имитационная модель предоставляет новые возможности для коллективной работы.

Во-первых, повышается её эффективность. В AnyLogic Cloud реализован очень простой принцип: в единой базе сохраняются входные параметры и соответствующие им результаты всех когда-либо проведённых запусков модели любым из пользователей. (При этом зерно генератора случайных чисел тоже считается входным параметром.) И, если параметры нового запуска найдены в базе, имитация не проводится (рис. 3).

Предположим, что вы когда-то проводили эксперимент с варьированием некоего параметра от 2 до 10. А ваш коллега хочет проверить интервал от 6 до 12. Тогда ему нужно будет сделать только два дополнительных прогона вместо семи – первые пять уже есть в базе. Даже если у вас нет коллег, и вы работаете индивидуально, вы всё равно получаете колоссальную экономию времени. Например, вы выполнили эксперимент Монте-Карло первого порядка со 100 репликациями и получили некое распределение на выходе. Вы хотите повысить точность и решаете, что вам нужно 150 репликаций. AnyLogic Cloud поймёт, что можно использовать 100 уже выполненных прогонов и, продолжая нужным образом генерировать зерно, дополнит их ещё 50. Если потом вам или кому-то ещё нужно будет проводить оптимизацию, некоторые пары (вход/выход) будут уже готовы, и так далее.

Во-вторых, и это касается в первую очередь открытых исследовательских и образовательных проектов, облако симитационными моделями, находящимися в открытом доступе (и желательно, с открытым кодом), представляет собой социальную сеть для исследователей, преподавателей и студентов с возможностью делиться наблюдениями и результатами, улучшать модели, оставлять комментарии и лайки. А для консультантов это готовая среда для размещения своего живого портфолио имитационных моделей.

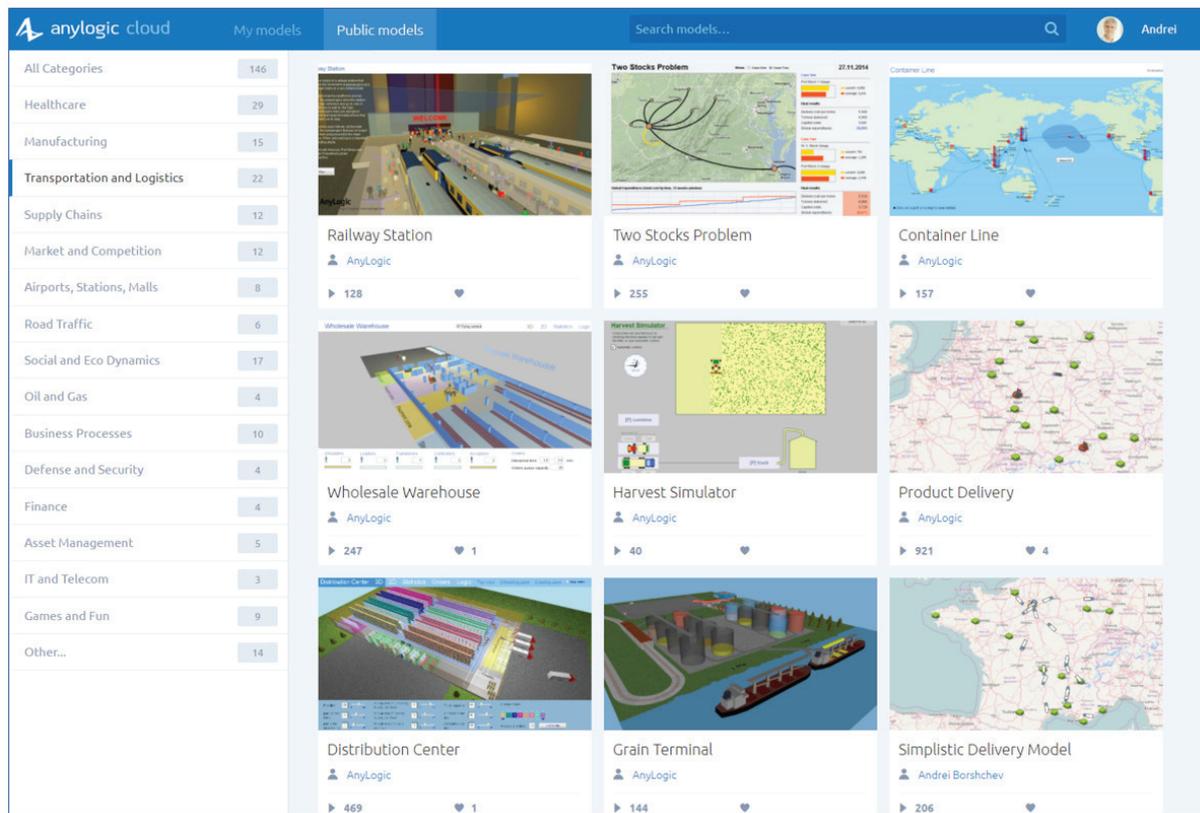


Рис. 7. AnyLogic Cloud – социальная сеть для ИМ

### Облако и вопросы безопасности данных

Проблема безопасности общая для всех облачных приложений. Существует ряд компаний и организаций, политика которых запрещает размещать данные в сторонних облачных сервисах (несмотря на то, что все они тоже защищены), но в то же время никому не хочется отказываться от их функциональности. Решением в этих случаях является так называемый *private cloud*, то есть облачное приложение, развёрнутое на частной инфраструктуре компании или специально арендуемой ею части облака провайдера.

### Миграция в облако и переход на подписку

Вместе с традиционными десктопными приложениями (а часто и быстрее их) отживает свой век и старая бизнес-модель, а именно, *permanent license* – лицензия без ограничения срока действия с опциональной покупкой поддержки. Из производителей инструментов ИМ никто еще не осмелился перейти на подписку для основного приложения (то есть редактора моделей). Но облачные добавки к нему, естественно, продаются только по *subscription*.

Облачный сервис для ИМ пока реализован у двух продуктов: AnyLogic [5] и Simul8 [6] (у Simio он декларирован, но на настоящий момент недоступен [7]). Тарифные планы предполагают бесплатную доступность ограниченной части сервиса для лицензированных пользователей самих продуктов (у AnyLogic и Simul8) или для всех вообще (AnyLogic). Далее различные уровни платной подписки в зависимости от интенсивности и сценариев использования облака. Эти планы, безусловно, будут меняться, так как компании пока находятся в стадии тестирования спроса.

### Заключение

Итак, доступность облачных сервисов существенно изменит сценарии использования готовых моделей, хотя и не окажет большого влияния на процесс их разработки. Можно сказать, что имитационная модель, выходя из рук разработчика, полностью отделяется от десктопного приложения, где она была создана, и продолжает свою независимую жизнь в облаке. Планирование и запуск экспериментов, просмотр анимации, анализ результатов становятся доступными в любой точке, где есть интернет и с любого устройства, где есть браузер, включая планшеты и телефоны. Упрощается и переходит в онлайн процесс итераций имитационных моделей с заказчиком и передачи результатов проекта. Выходит на новый уровень коллективная работа исследовательских и образовательных коллективов с моделями, которые находятся в публичном пространстве, возникает подобие социальной сети для ИМ. И, естественно, многократно сокращается время получения результатов многопрогонных экспериментов, таких как Монте-Карло, оптимизация, варьирование параметров. Лидеры рынка продуктов ИМ уже предлагают облачные решения как дополнение к средствам разработки, доступные по подписке.

### Литература

1. **А.В. Борщев.** Имитационное моделирование: состояние области на 2015 год, тенденции и прогноз // Труды Конференции ИММОД 2015, Москва. С. 14–22.
2. **A.V. Borshchev, and A.E. Filippov.** From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. The 22nd International Conference of the System Dynamics Society, 2004 (ISDC'04), Oxford, UK.
3. Insight Maker, <https://insightmaker.com/>.
4. GitHub, <https://github.com/>.
5. AnyLogic Cloud, <https://cloud.anylogic.com/#/landing>.
6. SIMUL8 Studio, <https://www.simul8.com/products/studio/>.
7. Simio Portal Edition, <https://www.simio.com/products/simio-portal.php>.
8. Amazon Web Services, <https://aws.amazon.com>.
9. Simulation in the Cloud is Becoming Mainstream, <http://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/6356/Simulation-in-the-Cloud-is-Becoming-Mainstream.aspx>.