

Статья посвящена проблемам изучения интерактивной культуры. Компьютерные программы и системы (*software*) заменили разнообразными механические и электрические технологии, использовавшиеся в XX веке для создания, хранения, распространения и взаимодействия с артефактами медиальной культуры. Программы стали нашим интерфейсом для коммуникации с миром, Другими, нашей памятью и нашим воображением — универсальным языком, на котором говорит мир, и его универсальным двигателем. Программное обеспечение в XXI веке является тем же, чем были двигатель внутреннего сгорания и электричество в веке XX. Данная статья обобщает некоторые из идей, высказанные автором в недавней книге *Software Takes Command* (2013) — первой попытке теоретического и исторического осмысления программного обеспечения в сфере медиа и его влияния как на медиапрактику, так и на саму сущность медиа. Тема

статьи находится на передовой линии современных медиаисследований, направленных на анализ интерактивных процессов в пространстве между пользователями и программами. Автор обращается к вопросу о том, как парадигмы *Digital Humanities* и *big data* могут использоваться для исследования культурной деятельности современных компьютерных пользователей (дизайнеров, геймеров или простых потребителей). В фокусе его внимания — не код программ или контент медиафайлов, но сам интерактивный временной опыт пользователей. Целью подобных исследований является извлечение и анализ разных паттернов, характеризующих интерактивные процессы. Если традиционная теория описывала структуру текста (*semiotics*) или восприятие некоего воображаемого читателя, то сегодня мы можем анализировать конкретные данные о поведении миллионов людей посредством программ, взаимодействующих с интерактивными текстами.

ваний)², который совместно с другими организациями в 2008 году запустил программу «Эффективные вычисления в гуманитарных науках», в 2009-м и 2011-м — конкурс «Погружаясь в данные»³, а также благодаря нескольким новаторским проектам отдельных гуманитарных исследователей и их лабораторий.

В статье *The New York Times* от 16 ноября 2010 года (в цикле *Humanities 2.0*) смело заявляется: «Следующий прорыв в языке, истории и искусстве? Данные»⁴. И хотя оцифровка историче-

2. *Digital Humanities Office at National Endowment of Humanities (National Funding Agency for Humanities Research)*.

3. *Digging into Data Challenge* (далее — «Погружаясь в данные») — конкурс международных исследовательских проектов на тему влияния и перспектив применения технологий «больших данных» и компьютерного анализа в целом в социальных и гуманитарных науках. Конкурс организуется и спонсируется исследовательскими фондами США, Великобритании, Нидерландов и Канады. Победившие проекты получают грантовую поддержку. Всего проведено 3 раунда конкурса: в 2009, 2011 и 2013 годах. — *Прим. пер.*

4. *Foys M. K. Digital Keys for Unlocking the Humanities' Riches // The New*

ских культурных артефактов вроде бы производит такие данные, представление самой *культурной деятельности* как данных, которые можно анализировать с помощью компьютера, — это вызов.

До сих пор все проекты с большими массивами данных в области цифровых гуманитарных наук, о которых я знаю, работали с *оцифрованными культурными артефактами* из прошлого. Если же применить парадигму больших данных к исследованию современных *интерактивных программных медиа*, то мы столкнемся с рядом интересных теоретических вопросов и вызовов. Что такое «большие данные» в случае интерактивных медиа? Каким могло бы быть исследование интерактивного временного опыта пользователей в отличие от анализа кода программ и контента медиафайлов?

Настоящая статья дает возможные ответы на эти вопросы. Они основаны на исследованиях в рамках проекта *Software Studies Initiative*, который я веду в Калифорнийском университете в Сан-Диего (*UCSD*) и в Калифорнийском институте телекоммуникаций и информации (*Calit2*), а также на результатах нашего текущего проекта, поддержанного грантом Национального научного фонда (*NSF*), в котором мы анализируем записи взаимодействий большого количества людей с огромным виртуальным миром *Scalable City*⁵.

Проект *Software Studies Initiative* был запущен в 2007 году для работы в рамках двух взаимосвязанных программ:

- 1) изучение ПО и его использования в современных обществах с помощью методов гуманитарных и социальных наук и теории новых медиа;
- 2) исследование культурных содержаний и процессов с помощью основанных на ПО методов. Иными словами, ПО является для нас и предметом исследования, и интеллектуальной технологией, которая добавляется к арсеналу других имеющихся гуманитарных методов чтения культуры.

Первая точка зрения — часть повестки «исследований ПО» (*Software Studies*) в их обычном понимании⁶, вторая же относит-

York Times. November 16, 2010. URL: <http://www.nytimes.com/2010/11/17/arts/17digital.html>.

5. См.: URL: <http://www.scalablecity.net/>.

6. См. предисловие к книжной серии «Исследования ПО» издательства *MIT Press*. URL: <http://mitpress.mit.edu/catalog/browse/browse.asp?type=6&serid=179>.

ся к «цифровым гуманитарным наукам» (*Digital Humanities*)⁷. Для обозначения последних мы используем отдельный термин — «культурная аналитика», поскольку в понятие «цифровые гуманитарные науки» ныне включают все виды цифровой работы в гуманитаристике: от кураторства онлайн-коллекций до вычислительного анализа и визуализации больших массивов культурной информации.

Исследования новых медиа (включая недавно появившиеся в их составе исследования ПО) за двадцать с лишним лет существования произвели тысячи книг, конференций и десятки тысяч статей, в которых были проанализированы многие аспекты интерактивных медиа⁸. Как бы то ни было, если мы хотим использовать парадигму больших данных для исследования интерактивных медиа, нам нужно теоретически рассмотреть этот тип медиа с дополнительных точек зрения. Такому анализу и посвящена первая часть данной статьи. Вторая же, опираясь на результаты первой, предлагает методологию для исследования интерактивных медиа как «больших массивов данных»⁹.

Что такое «данные» в интерактивных медиа?

Использование ПО переопределяет большинство социальных и культурных практик и заставляет нас переосмыслить понятия и теории, созданные нами для их описания. Возьмем, к примеру, современный «атом» культурного творчества, передачи и памяти — «документ»: некое содержание, хранимое в физической форме и поставляемое потребителю с помощью физических копий (книг, фильмов, аудиозаписей) или электронной трансляции (телевидения). В программной культуре больше нет бывших в ходу в XX веке «документов», «работ», «сообщений» или «записей». Вместо фиксированных документов, чье содержание и смысл можно было определить, изучив их структуру и оглав-

7. About Software Studies//SoftwareStudies.com. May 2007. URL: <http://lab.softwarestudies.com/2007/05/about-software-studies-ucsd.html>.

8. У одного только издательства *MIT Press* в категории «новые медиа» сейчас 368 книг. См. URL: <http://mitpress.mit.edu/disciplines/new-media-and-digital-humanities>.

9. В этой статье я развиваю идеи, которые впервые были озвучены в конкурсном предложении «Погружаясь в исходно цифровые данные: методы анализа интерактивных медиа», подготовленном в 2009 году мной, Джереми Дугласом, Мэтью Фуллером и Ольгой Горюновой для конкурса «Погружаясь в данные». Текст данной статьи основан на введении к книге: *Manovich L. Software Takes Command*. L.: Continuum, 2013.

ление (типичный для культурного анализа и культурной теории XX века ход, от русского формализма до литературного дарвинизма¹⁰), теперь мы взаимодействуем с динамическими «программными процессами».

Я использовал слово «процесс», поскольку переживаемое нами в опыте конструируется посредством ПО в режиме реального времени. Так что независимо от того, исследуем мы динамический веб-сайт, играем в видеоигру или используем приложение на мобильном телефоне, чтобы определить местонахождение конкретных мест или друзей поблизости, во всех этих случаях мы взаимодействуем не с заранее определенными статичными документами, а с динамическими продуктами вычислительного процесса, в режиме реального времени происходящего в нашем устройстве и/или на сервере.

Для создания этих процессов компьютерные программы могут использовать различные компоненты: шаблоны дизайна, хранящиеся на локальной машине файлы, медиа из баз данных на сетевом сервере, входной сигнал от мыши, сенсорного экрана, джойстика, наших движущихся тел или иного интерфейса и т. д. Следовательно, хотя какие-то статичные документы и могут использоваться, итоговый *опыт медиа, конструируемый ПО, обычно не соответствует никакому статичному документу, хранящемуся на некоем носителе (media)*. Другими словами, в противоположность рисункам, литературным трудам, нотным записям, фильмам, промышленным схемам или заданиям критик не может просто обратиться к единому «файлу», который бы вмещал все содержание работы.

Даже в таких, казалось бы, простых случаях, как просмотр документа в формате *pdf* или открытие фото в медиаплеере, мы уже имеем дело с «программными процессами», ведь именно программа задает возможность навигации, редактирования и пересылки документа, а не сам документ. Следовательно, изучая *pdf*- или *jpeg*-файл так, как в XX веке критики изучали роман, фильм или телевизионную передачу, мы бы узнали кое-что об опыте, получаемом нами при взаимодействии с этим документом посредством ПО, — что-то, но не все. Этот опыт в равной мере формируется интерфейсом и инструментами программ. Вот почему, если мы хотим осмыслить современные медиа, нам важно изучать инструменты, интерфейсы, допущения, понятия

10. Darwinian literary studies // Wikipedia.org. URL: http://en.wikipedia.org/wiki/Darwinian_literary_studies.

и историю культуры ПО (*Cultural Software*), включая теории его изобретателей, которые в 1960–1970-е годы определили большинство этих понятий.

Изменение природы того, что конституирует «медиадокумент», ставит под вопрос устоявшиеся культурные теории, зависящие от этого понятия. Рассмотрим интеллектуальную парадигму, преобладавшую в исследованиях медиа с 1950-х годов, — разработанный в рамках исследований коммуникаций взгляд на культуру как «передачу». Исследователи коммуникаций позаимствовали и применили к массмедиа базовую модель передачи информации, сформулированную Клодом Шенноном в статье 1948 года «Математическая теория связи»¹¹ и опубликованной следом в 1949 году книге¹² в соавторстве с Уорреном Уивером. В этой парадигме массовая коммуникация (иногда и культура в целом) описывалась как процесс коммуникации между авторами, создающими и отправляющими «сообщения», и адресатами (*audiences*), которые их «получают». Согласно этой парадигме, сообщения не всегда полностью декодируются получателями по техническим (шумы при передаче) или семантическим (непонимание заложенных смыслов) причинам.

Классическая теория коммуникации и индустрия медиа считали эту частичность приема проблемой; другую позицию занимал основатель британских культурных исследований Стюарт Холл. В своей влиятельной статье 1980 года «Кодирование/декодирование»¹³ он утверждал, что это позитивный феномен. Холл предположил, что адресат конструирует собственные смыслы на основе получаемой информации. Новые смыслы — скорее активная переинтерпретация посланных сообщений, нежели следствие неудачи коммуникации. Но и классические исследования коммуникации, и культурные исследования неявно принимали как само собой разумеющееся, что сообщение есть нечто завершенное и определенное — независимо от того, хранилось ли оно на физическом носителе (к примеру, на магнитной ленте) или создавалось в режиме реального времени отправителем (прямой эфир телевидения). Таким образом, предполагалось, что получатель коммуникации прочитает весь рекламный текст, посмо-

11. *Shannon C. E.* A Mathematical Theory of Communication // *Bell System Technical Journal*. 1948. Vol. 27. P. 379–423, 623–656.

12. *Shannon C. E., Weaver W.* The Mathematical Theory of Communication. Urbana, IL: University of Illinois Press, 1949.

13. *Hall S.* Encoding/decoding // *Culture, Media, Language/Centre for Contemporary Cultural Studies* (ed.). L.: Hutchinson, 1980.

трит весь фильм или послушает всю песню и только после этого верно или неверно проинтерпретирует полученное, припишет ему собственные смыслы, присвоит, перетасует и т.д.

Такое допущение попросту неприменимо к интерактивным программным медиа, хотя оно и ставилось под сомнение уже в связи с появлением в 1999 году цифровой видеозаписи (*DVR*), которая породила феномен «смещения времени». Интерфейсы обеспечивающих доступ к медиа приложений, таких как веб-браузеры и поисковые машины, архитектура гиперссылок *www* и интерфейсы онлайн-медиа-сервисов, предлагающих множество медиаартефактов для игры, предпросмотра и/или покупки (*Amazon, Google Play, iTunes, Rhapsody, Netflix* и т.д.), побуждают людей «пролистывать» страницы, быстро перемещаясь между медиа одновременно горизонтально (от одного результата поиска к следующему, от одной песни к другой и т.д.) и вертикально (*по структуре* медиаартефактов, к примеру от списка треков музыкального диска к конкретному треку). Кроме того, благодаря им теперь можно легко начинать и заканчивать смотреть/проигрывать медиа в произвольной точке. Иными словами, «получаемое» пользователем «сообщение» не только активно «конструируется» им (посредством когнитивной интерпретации), но и активно управляется (он определяет, какую информацию и как получать).

Не менее важно и то, что, когда пользователь взаимодействует с приложением, которое предоставляет медиаконтент, этот контент часто не имеет каких-либо определенных, конечных границ. К примеру, пользователь *Google Earth* каждый раз, используя приложение, будет переживать в опыте разную «Землю». Дело в том, что *Google* мог обновить какие-то спутниковые фотографии или добавить панорамы новых улиц. Кроме того, могли быть добавлены новые трехмерные здания, новые слои и новая информация об уже существующих уровнях. Более того, пользователь в любой момент может загрузить еще геопространственную информацию, созданную другими пользователями и компаниями, либо выбрав одну из опций в меню *Add* (интерфейс *Google Earth 6.2.1*), либо напрямую открыв *KLM*-файл. *Google Earth* — хрестоматийный пример нового типа медиа, ставшего возможным благодаря сети: это интерактивный «документ», чье содержание целиком не задано заранее. Оно меняется и растет с течением времени.

В некоторых случаях это может и не иметь особого значения для более крупных «сообщений», передаваемых в коммуникации

программными приложениями, веб-сервисами, играми и иными типами интерактивных медиа. Например, в случае *Google Earth* включенные уровни и новый контент, который был добавлен пользователями с момента вашего последнего посещения, не влияют ни на одну из встроенных в приложение норм обозначений — на представление территорий с помощью перспективной проекции (*GPP*, метод картографической проекции¹⁴).

Однако *Google Earth* — не просто «сообщение», поскольку помимо прочего пользователь может добавлять собственные медиаинструменты и информацию к базовой репрезентации, предоставляемой приложением, тем самым создавая сложные и насыщенные с точки зрения медиа проекты на основе имеющейся геоинформации. Это *платформа*, на которой пользователи могут конструировать что-то. И хотя здесь можно усмотреть некоторую преемственность с тем, как в XX веке пользователи творчески перерабатывали коммерческие медиа (примеры — поп-арт и апроприация в искусстве, музыкальные ремиксы, слэши и видео¹⁵ и т. д.), различия гораздо значительнее, чем сходства.

Переход от сообщений к платформам был в центре трансформации сети, которая получила название *Web 2.0* в 2004–2006 годах. Веб-сайты 1990-х, предоставлявшие контент, произведенный другими (и потому передававшие «сообщение»), были дополнены социальными сетями и сайтами с социальными медиа, где пользователи могут делиться данными, комментировать и тегировать собственные медиа. Статья в «Википедии», посвященная *Web 2.0*, описывает эти различия следующим образом:

Сайт *Web 2.0* позволяет пользователям взаимодействовать и сотрудничать друг с другом в рамках социальных медиа в качестве создателей (просьюмеров) пользовательского контента в виртуальном сообществе — в противоположность веб-сайтам, на которых потребители ограничены пассивным просмотром созданного для них контента. К *Web 2.0* относятся социальные сети, блоги, вики-проекты, арендуемые сервисы администрирования, веб-приложения, мэшапы и фолксономии¹⁶.

К примеру, если снова обратиться к *Google Earth*, пользователи добавляли разнообразную информацию о глобальной ситуа-

14. См.: General Perspective Projection//Wikipedia.org. URL: http://en.wikipedia.org/wiki/General_Perspective_projection.

15. См., напр.: Penley C. *Feminism, Psychoanalysis and the Study of Popular Culture*//Cultural Studies/L.Grossberg (ed.). L.: Routledge, 1992.

16. См.: Веб 2.0//Wikipedia.org. URL: http://ru.wikipedia.org/wiki/Веб_2.0.

ции (*Global Awareness Information*), включая информацию о сертификации справедливой торговли, данные Гринписа и Цели развития тысячелетия (ООН)¹⁷. Либо, например, вы можете встроить *Google Maps*, *Wikipedia* или контент, предоставляемый большинством других крупных сайтов *Web 2.0*, напрямую в свой мэшап — еще более непосредственный способ взять предоставляемый веб-сервисами контент и использовать его для создания собственной платформы.

Широкое распространение сервисов *Web 2.0* и разнообразных веб-инструментов коммуникации (онлайн-форумы, посвященные обсуждению популярных ПО, коллективное редактирование в «Википедии», твиттер и т.д.) обеспечивает быстрое распознавание пропуска данных, отбора, цензуры и иных типов «плохого поведения» издателями ПО — еще одна черта, отличающая распространяемый веб-компаниями контент от массовых медиа XX века. К примеру, каждая статья на «Википедии», посвященная какому-нибудь сервису *Web 2.0*, содержит специальный раздел, посвященный дискуссиям, критике или ошибкам.

Во многих случаях можно использовать эквиваленты платных и закрытых приложений с открытыми исходными кодами. ПО с открытыми исходниками и/или бесплатное ПО (не всякое бесплатное ПО предполагает открытые исходники) часто дают дополнительные возможности создания, комбинирования и распространения как контента, так и новых программных дополнений. Это не значит, что ПО с открытыми исходниками всегда использует другие, нежели коммерческое ПО, допущения и технологии. Например, можно выбирать между альтернативами *Google Earth* и *Google Maps — OpenStreetMap*, *Geocommons*, *WorldMap* и др., каждая из которых имеет либо открытые исходные коды, либо лицензии свободного программного обеспечения¹⁸. Примечательно, что коммерческие компании зачастую используют данные, полученные от таких бесплатных, коллективно разрабатываемых систем, поскольку в них содержится больше информации, чем в собственных системах компании. Например, *OpenStreet Map*, имевшая к началу 2011 года 340 тысяч участников¹⁹, используется *Flickr* и *Four-*

17. См.: Google Планета Земля//Wikipedia.org. URL: http://ru.wikipedia.org/wiki/Google_Планета_Земля.

18. См. URLs: <http://geocommons.com/>; <http://worldmap.harvard.edu>.

19. См.: Counter-mapping//Wikipedia.org. URL: <http://en.wikipedia.org/wiki/Counter-mapping>.

*square*²⁰. Кроме того, у пользователя есть возможность посмотреть код ПО с открытыми исходниками, чтобы лучше понять лежащие в его основе допущения и технологии.

Непрерывно меняющийся и растущий контент веб-сервисов и сайтов, разнообразие механизмов навигации и взаимодействия, возможность добавлять собственные содержания и объединять контент из различных источников, наличие архитектуры для совместных разработок и редактирования, а также механизмов мониторинга провайдеров — вот что решительно отделяет интерактивные сетевые программные медиа от медиадокументов XX века. Даже когда пользователь работает с одним локальным медиадокументом, который хранится в одном компьютерном файле (довольно редкая ныне ситуация), такой документ, опосредованный программным интерфейсом, отличается от медиадокументов XX века. Опыт пользователя все еще лишь отчасти определяется содержанием и организацией файла. Пользователь может произвольно перемещаться по документу, выбирая, какую именно информацию в какой последовательности просмотреть. И хотя «старые медиа» (за исключением теле- и радиовещания в XX веке) тоже предоставляли такую произвольность обращения, интерфейсы программных медиаплееров и приложений просмотра дают больше дополнительных возможностей для просмотра медиа и выбора того, к чему и когда обращаться.

К примеру, *Adobe Acrobat* может отображать эскизы каждой страницы *pdf*-документа; *Google Earth* может быстро масштабировать картинку заданной местности; онлайн-цифровые библиотеки, базы данных и хранилища (репозитории), содержащие научные статьи и абстракты (к примеру, *ACM Digital Library*, *IEEE Xplore*, *PubMed*, *Science Direct*, *SciVerse Scopus* и *Web of Science*), показывают тексты, в которых есть ссылки на выбранную вами статью²¹. Что важнее, эти новые инструменты и интерфейсы реализованы не на аппаратном уровне самих медиадокументов (как в случае возможности произвольного обращения к страницам печатной книги) или механизмов доступа к медиа (радио). Они являются частью отдельного программно-го уровня. Такая архитектура медиа позволяет легко добавлять

20. См.: List of Academic Databases and Search Engines//Wikipedia.org. URL: http://en.wikipedia.org/wiki/List_of_academic_databases_and_search_engines.

21. См.: Ibidem.

новую навигацию и новые инструменты управления, не изменяя сами документы. Например, я могу одним кликом добавить кнопку «Поделиться» в свой блог, обеспечив тем самым новые траектории циркуляции своего контента. Открыв текстовый документ в *Mac OS Preview*, я могу выделять цветом, добавлять комментарии и ссылки, рисовать и добавлять заметки. *Photoshop* позволяет мне сохранять свои правки в отдельных «корректирующих слоях», не меняя оригинальную картинку. И так далее и тому подобное.

Как следовать за пользователями программ

Исследование новых свойств интерактивных программных сетевых медиа требует также новых методов и новых теоретических концептов.

1.

Нужно иметь возможность записывать и анализировать *интерактивный опыт*, то есть конкретные временные взаимодействия определенных пользователей с ПО — в противоположность анализу исключительно «медиадокументов» (элементов, участвующих в конструировании этого опыта). К примеру, мы должны следовать за перемещением пользователей по сайту, а вовсе не ограничиваться изучением его контента. Мы должны следовать за игроками по мере их продвижения в видеоигре, не ограничивая анализ исследованием геймплея. Мы должны следовать за посетителями интерактивных инсталляций, изучающими пространство возможностей, заданное дизайнером, — возможностей, которые становятся актуальными событиями, лишь когда посетители действуют в соответствии с ними²².

Почему это так важно? В чем состоит разница между регистрацией процесса взаимодействия пользователя с традиционными медиа и использованием таких записей для анализа, с одной стороны, и теми же операциями с программными ме-

22. Здесь я провожу параллель с поворотом в исследованиях науки, произведенным Бруно Латуром и его коллегами, когда они сместили фокус внимания с исследования научных документов на следование за учеными в их лабораториях. См., напр.: *Латур Б.* Наука в действии: следуя за учеными и инженерами внутри общества. СПб.: Издательство Европейского университета, 2013. Сходным образом мы должны следовать за пользователями в их взаимодействиях с ПО, а не анализировать медиадокументы сами по себе.

диа — с другой? В конце концов, мы ведь можем использовать разнообразные технологии, чтобы фиксировать следы эмоциональных и когнитивных состояний субъекта, вызванных чтением книги, просмотром фильма, перемещением по антропогенной среде или взаимодействием с другими, более старыми формами медиа. Мы можем записывать показатели активности мозга, используя МРТ или ЭЭГ, записывать движения глаза, отслеживать пульс, реакцию кожи на раздражители, а также задействовать другие биометрические техники. К 2012 году ряд нейромаркетинговых компаний уже работали с голливудскими студиями, снимая такие показатели со зрителей тестовой аудитории и помогая режиссерам улучшать монтаж художественных фильмов и размещение скрытой рекламы²³ — новая услуга, выросшая из академических исследований «нейрокинематики»²⁴.

Но совсем другое дело, когда мы записываем и анализируем движения курсора по экрану, пространственную траекторию контролируемого пользователем персонажа игры или иные элементы опыта взаимодействия с программными медиа. Во всех этих случаях мы фиксируем и анализируем актуальную «работу», как она *совместно создается ПО и взаимодействиями с ним пользователя*, — работу, которая возникает лишь в течение этого взаимодействия и меняется в зависимости от сессии и пользователя. Это отличается от записи биометрических данных тестовой аудитории фильма, который был создан заранее.

Хотя все виды культурного опыта можно описать как формы взаимодействия, программные интерактивные медиа представляют собой новую форму человеческой культуры. Конечно, о читателе неинтерактивного текста тоже можно сказать, что он когнитивно конструирует свою собственную версию текста. Однако на материальном уровне «входные данные» (то есть текст на странице, фильм, показанный в кинотеатре, интерьер здания, форма продукта и т.д.) одни и те же для всех пользователей (по крайней мере, пока). В случае же взаимодействия с ПО эти «входные данные» изменяются при каждом его использова-

23. См.: *Randall K. Rise of Neurocinema: How Hollywood Studios Harness Your Brainwaves to Win Oscars* // *FastCompany.com*. February 25, 2011. URL: <http://www.fastcompany.com/1731055/rise-neurocinema-how-hollywood-studios-harness-your-brainwaves-win-oscars>.

24. *Hasson U., Landesman O., Knappmeyer B., Vallines I., Rubin N., Heeger D.J. Neurocinematics: The Neuroscience of Film* // *Projections*. Summer 2008. Vol. 2. Iss. 1. P. 1–26. URL: http://www.cns.nyu.edu/~nava/My-Pubs/Hasson-etal_NeuroCinematics2008.pdf.

нии. Они совместно создаются ПО и моим поведением в момент взаимодействия.

К примеру, видеоигра, в которую вы будете играть, скорее всего, будет отличаться от игры, в которую буду играть я. Почему? Потому что вероятность того, что другой игрок пройдет уровень по тем же пространственным траекториям, выполняя все квесты в той же последовательности и в точности повторяя все остальные действия, близка к нулю. Вариативность на самом деле еще выше, если учесть, что для управления производством объектов и персонажей и их действиями в играх используется генератор случайных чисел.

Аналогичным образом содержание главной страницы *Amazon.com* будет меняться при каждом ее посещении, поскольку ПО сайта автоматически настраивает функциональные части контента страницы (раздел *Your Recent History*, рекомендации и прочее) в зависимости от данных о моей истории просмотра и покупок, а также данных о поведении людей, сделавших схожие выборы. И хотя в момент написания этой статьи интерфейсы компьютеров и мобильных устройств не меняются от сессии к сессии — программы не переупорядочивают себя на рабочем столе, а иконки документов не меняют свой цвет в зависимости от частоты своего использования, — вполне допустимо наделить их большей «интеллектуальностью». Исследователи в области «дополненного познания» разрабатывают интерфейсы и среды обучения, которые постоянно менялись бы на основании входных данных от сенсоров, регистрирующих телесную и мозговую активность человека²⁵.

Любой анализ программных интерактивных медиа должен учитывать лежащее в их основе совместное творение (пользователями и программой) как их генеалогическую черту и встроенную в них вариативность. Но если мы не можем просто анализировать медиадokument, то как схватить множественные «программные действия» и опыт пользователей? Иными словами, *как нам следовать за пользователями?*

Культура всегда включала в себя взаимодействия между публикой и культурными объектами или между исполнителями и участниками. Однако до недавнего времени записывать и анализировать эти взаимодействия было трудно и затратно. Только голливудские киностудии и большие рекламные агентства могли позволить себе специальное оборудование, с помо-

25. См. URL: <http://www.augmentedcognition.org/>.

щью которого можно было записать простые эмоциональные реакции зрителей из заданной выборки на рекламу или концовку фильма и объединить эти данные с результатами других техник исследования фокус-групп (интервью, вопросников и т.д.). Как уже упоминалось выше, теперь эмоции можно фиксировать с помощью ЭЭГ, МРТ и иных биометрических техник, но большинство из них все еще дороги, и их использование и интерпретация результатов требуют специального образования. Как бы то ни было, отслеживание взаимодействий пользователей с самими ПО тривиально.

Почему? Дело в том, что сам феномен взаимодействия возможен благодаря тому, что программа постоянно отслеживает входные сигналы от интерфейса. Без этого не было бы никакого взаимодействия. Программное приложение постоянно фиксирует нажатия клавиш, движения мыши, выбор пунктов в меню, скольжение пальцев по тачпаду, голосовые команды и иные типы входных данных. Эти фиксируемые входные сигналы запускают предусмотренные приложением операции — масштабирование локации на карте, огонь из оружия в игре, изменение картинки с помощью инструмента «Кисть», поиск веб-страниц в соответствии с запросом пользователя и т.д.

Используемые сегодня всеми приложениями графические пользовательские интерфейсы (*GUI*) требуют фиксации множества типов вводных данных от пользователя и по иной причине. Если вы используете *UNIX*, то знаете, что после ввода команды часто не получаете никакой обратной связи (кроме случаев использования некорректного синтаксиса команды — тогда вы получаете сообщение об ошибке). Аналогично правка в *HTML*-документе не отображается до предварительного просмотра документа в браузере. Но в программах, использующих *GUI*, все входящие сигналы от пользователя эксплицитно подтверждаются. Компьютер дает немедленный текстовый, визуальный и/или звуковой ответ пользователю.

Если я редактирую документ, программа сразу показывает внесенные мной изменения. Пока я печатаю это предложение, каждый раз, когда я нажимаю клавишу, текстовый редактор фиксирует нажатие этой клавиши и немедленно обновляет отображаемый на экране документ. Можно сказать, что этот интерфейс имитирует печатную машинку XIX века, в которой нажатие клавиши вызывало механическое движение, посредством которого на бумаге печаталась соответствующая буква. Другой пример фиксации и хранения интерфейсами входных сигнала-

лов от пользователя — это окно «История» (или команда *history* в *UNIX*), показывающее последовательность всех введенных в течение сеанса пользователя команд.

Поскольку фиксация входных сигналов от пользователя уже является фундаментальным принципом современных интерактивных программных интерфейсов, ПО можно легко модифицировать, чтобы хранить значения этих входящих, а не стирать их, как только действие запущено. Это так же легко, как запись нового значения переменной в файл — действие, которое можно добавить с помощью одной строчки кода. Хотя запись и хранение этих сигналов в принципе были возможны еще с ранних этапов программной культуры, переход от настольного компьютера к веб-информатике в 1990-е годы сделал эту возможность фундаментальным элементом «программно-медийного комплекса». Поскольку динамические веб-сайты и сервисы управляются программами, находящимися на серверах компании, можно легко протоколировать детали их взаимодействий. К примеру, каждый веб-сервер хранит подробную информацию обо всех визитах на данный сайт. Чтобы помочь пользователям анализировать эту информацию и настраивать веб-дизайн своего сайта, была создана отдельная категория ПО и сервисов вроде *Google Analytics*.

Вслед за *Google Analytics*, запущенным в 2005 году, многие компании, занимающиеся социальными медиа, открывают пользователям доступ к части записываемой информации о взаимодействиях посетителей с сайтом, блогом или аккаунтом пользователя. Кроме того, они предоставляют пользователям интерактивные визуализации, чтобы помочь выявить паттерны и сделать их предложения в сети успешнее. Я могу работать с персональной аналитикой своего сайта, блога нашей лаборатории *Software-Studies.com*, нашего аккаунта в *Tube* и *Flickr*, страницы в *Facebook*, могу увидеть оценку популярности любого поста в блоге, фото, видео и обновления. Кроме того, компании делают общедоступной такую информацию по всем страницам или аккаунтам в *сети*. К примеру, в *YouTube* статистика есть под каждым видео, в ней графически отображаются совокупное количество просмотров, важные рекомендации и основные демографические показатели аудитории видео. Однако остальная фиксируемая информация остается в частном доступе и используется для улучшения алгоритмов, подбора рекомендаций и продажи рекламы²⁶.

26. См.: Social Search // Wikipedia.org. URL: http://en.wikipedia.org/wiki/Social_search; Рекомендательная система // Ibidem. URL: <http://ru.wiki->

Игровые компании отслеживают множество деталей геймплея своих многопользовательских онлайн-игр (ММО); затем эта информация используется для улучшения игрового дизайна. При разработке новых игр компании также систематически записывают и анализируют игру десятков тестирующих продукт пользователей («аналитика игр»). Другой пример того, как запись и использование информации о взаимодействиях пользователей с ПО оказываются в самом сердце современных веб- и медиаиндустрий, — поисковые алгоритмы *Google*²⁷. Алгоритмы используют сотни переменных, чтобы определить наиболее релевантные конкретному поисковому запросу веб-страницы, и одним из типов входных данных является информация о том, на какие из ссылок, выданных ранее по тому же запросу, кликали пользователи.

Сайты социальных медиа и электронной торговли вроде *YouTube* и *Amazon* тоже регистрируют и анализируют входные данные от пользователя, снабжая ими свои рекомендательные системы. Как показали Франк Кесслер и Мирко Тобиас Шаффер, в случае *YouTube* операции, кажущиеся пользователю побочными по отношению к основной деятельности, то есть просмотру видео (тегирование, комментирование и сообщение о недопустимости некоторых видео), оказываются ключевыми для работы сервиса в целом:

Все эти операции, которые предлагаются пользователям *YouTube* (или, скорее, которые должны осуществляться, чтобы производились метаданные, необходимые для функционирования *YouTube*), на первый взгляд, являются вспомогательными опциями и дополнительными сервисами. Однако все наоборот: в действительности они обеспечивают необходимую основу для управления базой данных. <...> Фактически каждый клик по одной из ссылок на клип, каким бы случайным он ни был, попадает в базу данных²⁸.

Поскольку *YouTube* не раскрывает свои технологии, а его программисты на конференциях вдаются в подробности лишь отно-

pedia.org/wiki/Рекомендательная_система; Semantic Targeting // Ibid. URL: http://en.wikipedia.org/wiki/Semantic_targeting.

27. См. URL: <https://www.google.com/insidesearch/howsearchworks/algorithms.html>.

28. Kessler F., Schaffer M. T. Navigating YouTube: Constituting a Hybrid Information Management System // The YouTube Reader / J. Burgess, P. G. Lange, M. T. Schafer (eds). Stockholm: National Library of Sweden, 2009. P. 284–285.

сительно некоторых из них, я не могу проверить, верен ли этот анализ во всех своих аспектах²⁹. Нам также неизвестно, используются ли в его алгоритмах записываемые и сохраняемые данные обо всех типах пользователей. Наконец, записываемое хранится, конечно, не в одной базе. (Часть задачи исследований ПО состоит в том, чтобы побудить исследователей быть более точными в своих анализах ПО, так как все эти детали важны.) Как бы то ни было, в целом аргумент Кесслера и Шаффера верен — веб- и медиакомпании фиксируют и анализируют входные данные от пользователя, чтобы улучшать свои продукты и предложения, и эта информация очень важна для их бизнеса, даже если для пользователей это может быть не вполне очевидно.

Дисциплинам, связанным с изучением культуры, тоже стоит использовать эти типы данных (и конечно, вы должны сначала спрашивать пользователей, согласны ли они на запись и анализ этой информации). Исследователи слишком часто полагаются только на свои интуиции и личный опыт в связи с такими продуктами. Необходимость работать с действительными фактами о поведении множества конкретных пользователей может порядком освежать. На деле это означает, что вдобавок к теоретизированию о воображаемом абстрактном пользователе (типичная для гуманитариев стратегия), или использованию опросов с целью ознакомления с культурным опытом людей, или этнографии (исследованиям коммуникаций) — вдобавок ко всему этому мы можем собирать и анализировать данные о действительных взаимодействиях людей с программными артефактами.

Таким образом, мы сможем настроить культурный анализ на процессы культурного производства в эпоху программ (*Software Society*), используя те же методы, но для других целей. К примеру, дизайнеры программных приложений и контента собирают данные об опыте пользователей, *тестирующих* ПО, а затем используют эту информацию, чтобы улучшить свои продукты. Как культурные критики, мы можем собирать данные об опыте *реальных* пользователей, чтобы лучше понимать, что

29. Davidson J., Liebold B., Liu J., Nandy P., Vleet T. V., Gargi U., Gupta S., Yu He, Lambert M., Livingston B., Sampath D. The YouTube Video Recommendation System // Proceedings of the 4th ACM conference on Recommender Systems (RecSys '10). N.Y.: ACM, 2010. P. 293–296. DOI=10.1145/1864708.1864770; Zhou R., Khemmarat S., Gao L. The impact of YouTube Recommendation System on Video Views // Proceedings of the 10th Annual Conference on Internet Measurement (IMC '10). N.Y.: ACM, 2010. P. 404–410. DOI=10.1145/1879141.1879193.

это за опыт, картографировать его изменчивость и, что важнее, сравнивать опыт разных продуктов, проясняя тем самым культурные паттерны. Например, если команда аналитиков игровой компании обычно только анализирует игры, разрабатываемые исключительно этой компанией, то мы можем взглянуть шире, проанализировав геймплей многих игр (одного или нескольких жанров) или геймплей разных версий одной игры на протяжении десятилетий их эволюции³⁰. Если вас пугает предложение освоить программные техники, с помощью которых компании создают медиа, для анализа самих медиа, то простите, что втягиваю вас в XXI век...

Поскольку у нас обычно нет доступа к коду коммерческих медиапродуктов, можно использовать другие методы регистрации взаимодействий. Один из них — запись видео геймплея (он систематически использовался нами в лаборатории исследований ПО с 2008 года)³¹. Подход состоит в том, чтобы превращать каждую игровую сессию в эквивалент фильма, записывая видео с экрана игрока в течение игровой сессии. Этот процесс аналогичен использованию снимков экрана в обучающих видео для программных приложений. Создав такую запись, мы можем анализировать игровой опыт как процесс изменения с течением времени.

Можно графически представить ритмы и паттерны в рамках одной игровой сессии. Можно собрать данные многих сессий для визуализации пространства возможного опыта, задаваемого конкретной игрой. Далее можно создать визуальные «ландшафты» более крупных областей культуры видеоигр — с массивами данных, представляющих различные игровые сессии для одной игры, и сверхмассивами, представляющими сравнение многих игр. Наконец, если включить еще больше данных, то можно визуализировать исторические паттерны игровой культуры. Поскольку многие старые игры доступны в программной эмуляции, этот метод хорош как способ понять историю игр.

30. Бен Миддлер рассматривает историю записи геймплея и ее использования в своей статье: *Middler B. Generations of Game Analytics, Achievements and High Scores*//Eludamos. Journal for Computer Game Culture. 2009. Vol 3. № 2.

31. Примеры наших проектов, в которых используется этот метод, URLs: <http://lab.softwarestudies.com/2008/06/videogameplayviz-analyzing-temporal.html>; <http://lab.softwarestudies.com/2012/02/kingdom-hearts-game-play-visualizations.html>.

Обратите еще раз внимание на фундаментальную разницу между записью опыта интерактивных медиа и записью опыта статического медиадокумента (к примеру, регистрация движения глаз человека во время просмотра им видеорекламы). Запись изображения с экрана играющего в видеоигру пользователя схватывает не только *как нечто переживается*, но и конкретный медиаартефакт, *сконструированный пользователем и программой* в течение конкретной сессии, — иными словами, *объект опыта*.

Хотя сегодня исследователи видеоигр обычно основывают свой анализ на собственном взаимодействии с интерактивными медиа, есть веские основания полагать, что серия таких записей (представляющая множество сессий множества игроков) должна быть принята в качестве базовой «единицы» культурного анализа этого типа медиа. Можно распространить эту идею и на другие типы интерактивных медиа. Исследователь может совместить использование собственного опыта конкретного интерактивного артефакта с использованием записи множества сессий (в идеале представляющих как можно больше разных пользователей), тем самым получив репрезентативный образец пространства возможностей в целом, конституирующего «интерактивный текст».

2.

Особенности интерактивных медиа также ставят под вопрос значение термина «большие данные», установленное в исследованиях культуры. Уже упоминавшиеся грантовые программы, учрежденные в 2008–2009 годах в США Национальным фондом для Отдела гуманитарных наук для проектов в области цифровых гуманитарных наук, призывали гуманитарных ученых начать применять вычислительные техники для анализа крупномасштабных массивов информации, ставшего стандартом для других наук. Описание второго конкурса «Погружаясь в данные», совместно организованного *NEH* и *NSF*³², начиналось с таких вопросов:

Теперь, когда у нас есть большие базы данных материалов, используемых учеными в гуманитарных и социальных науках (от оцифрованных книг, газет и музыки до таких транзакционных данных, как результаты веб-поиска, инфор-

32. *National Endowment for the Humanities, National Science Foundation.*

мация сенсорных датчиков или записи сотовых телефонов), какие новые, основанные на вычислениях исследовательские методы могли бы мы освоить?³³

Парадигма больших массивов данных особенно значима для исследований интерактивных медиа. Благодаря проектам по оцифровке, осуществляемым библиотеками, музеями, организациями по сохранению культурного наследия и различными компаниями, у ученых теперь есть доступ к большому количеству культурных артефактов из «старых медиа» — миллионам страниц газет, книг, карт, исторических записей, фильмов и телепрограмм, — и их можно анализировать вычислительными методами. Впрочем, на фоне количества доступных для исследования цифровых культурных артефактов масштабы «больших данных» традиционных культурных архивов довольно невелики. Сравните для примера следующие цифры. К 2010 году *Google Books* отсканировал свыше 15 миллионов книг³⁴, *Europeana* предоставила доступ к 10 миллионам цифровых культурных объектов³⁵, а *ARTstor.org* (стандартный онлайн-ресурс для преподавания истории искусств) принял более 1 миллиона цифровых изображений произведений искусства, предоставленных музеями и коллекциями. Но если обратиться к интерактивным медиа, то здесь счет идет на миллиарды: к тому же к 2010 году *Archive.org* собрал 150 миллиардов веб-страниц (начиная с 1995 года), а пользователи *YouTube.com* ежеминутно загружали по 24 часа нового видео. А к началу 2012 года, по данным *Facebook*, его пользователи ежемесячно загружали по 7 миллиардов фотографий.

И это еще не все. Программная интерактивная культура не только значительно увеличивает масштаб количества создаваемых медиадокументов. Она также делает каждый из документов потенциально бесконечным по размеру. Другими словами, «массивы данных» цифрового опыта не просто велики — они бесконечны. Позвольте объяснить.

В случае неинтерактивных медиаартефактов «большие данные» отсылают к размеру — как к размеру необходимого им физического или цифрового хранилища, так и ко времени, требующемуся человеку, чтобы обработать эти артефакты (прочитать книги, просмотреть фильмы и т. д.). Следовательно, хотя оци-

33. См. URL: <http://www.diggingintodata.org>.

34. Crawford J. On the Future of Books. Пост в блоге *Google Books Search*. URL: <http://booksearch.blogspot.ru/2010/10/on-future-of-books.html>.

35. См. URL: <http://version1.europeana.edu/>.

фровка и создала большие массивы культурных данных, требующие вычислительных методов для своего анализа, эти массивы конечны. В случае же интерактивных цифровых медиа «большой» характер их данных требует переосмысления.

К примеру, мы хотим исследовать определенную видеоигру. Как мы уже видели, такая игра не соотносится ни с одним конкретным текстом — напротив, каждый раз, когда игрок вступает в игру, он производит новый геймплей. Умножьте это на количество игроков, и массив данных существенно возрастет. Исследование одной игры теперь предполагает анализ тысяч часов видео с полными игровыми сессиями множества игроков. Учитывая, что обычная однопользовательская игра рассчитана примерно на 40 часов прохождения, 10 полных сессий для 10 игроков уже дадут 4000 часов видео.

Таким образом, хотя игра, понятая как *текст* (или как *работа*), может быть довольно небольшой (программное устройство игры и ее медиаобъекты — это несколько гигабайт), та же игра в качестве *интерактивного опыта* (или *порождающего процесса*) потенциально бесконечна, поскольку каждая игровая сессия уникальна. Это значит, что мы начинаем систематически анализировать интерактивные медиа в терминах актуального участия пользователя и больше не работаем с просто *большими массивами данных*. Вместо этого мы вынуждены заняться *бесконечными массивами данных*.

3.

Поскольку «программные медиа» — это частный случай ПО в целом (то есть это одна или несколько компьютерных программ, написанных на некотором стандартном компьютерном или скриптовом языке, которые обычно используют набор медиаобъектов — тексты, трехмерные модели, изображения, графику, звуки — и входные сигналы пользователя, чтобы производить интерактивный опыт), то логично спросить, можем ли мы проанализировать и код этих программ. Однако, хотя существование кода отдельно от актуального опыта «интерактивного текста» — это одна из определяющих черт программной культуры, на деле исследование программ, реализованных программно, не так уж просто. Оно порождает собственные теоретические и методологические проблемы.

Ранние программы, такие как видеоигры 1970-х, были относительно короткими. Они представляют собой идеальные объекты для исследований кода. Однако в случае любых современных

коммерческих интерактивных медиапроектов, медиаприложений или операционных систем программный код будет попросту слишком длинным и сложным для осмысленного прочтения. К тому же придется исследовать все используемые им библиотеки кода.

Обратите внимание на следующие цифры. Если *Windows NT 3.1* (1993) по оценкам содержит 4–5 миллионов строчек кода, то в *Windows XP* (2003) уже 45 миллионов³⁶. *Mac OS*, известная своей элегантностью и простотой по сравнению с *Windows*, на самом деле даже больше — *OS X 10.4* (2006) содержит 86 миллионов строк. А как насчет *Adobe Creative Suite*? Предполагаемое количество строк в *Adobe CS3* — около 80 миллионов³⁷. (*Gmail* в этом отношении преуспел — в нем только 443 тысячи строк *Javascript*³⁸.)

Но размер программного кода — не единственная проблема. Если приложения и операционная система, работающие на одной машине, уже обладают потрясающей сложностью, то постепенное перемещение программных приложений в сеть порождает множество новых аспектов. Допустим, мы хотим проанализировать веб-сервис, веб-приложение вроде *Gmail* или динамический веб-сайт. В противоположность статичным веб-сайтам, поначалу преобладавшим в интернете, контент страницы динамического веб-сайта может меняться в зависимости от действий пользователя. *Javascript, Flash, PHP, Perl* и *CGI* — примеры технологий, используемых сегодня для создания таких сайтов³⁹.

Веб-сервисы, веб-приложения и динамические веб-сайты часто используют *многоуровневую программную архитектуру*, в которой разные программные модули (например, веб-клиент, сервер приложения и база данных⁴⁰) работают вместе. В особенности в случае масштабных коммерческих динамических сайтов, к примеру *amazon.com*, то, что пользователь воспринимает как единую веб-страницу, оказывается узлом непрерывных

36. См.: Количество строк кода//Wikipedia.org. URL: http://ru.wikipedia.org/wiki/Количество_строк_кода.

37. См. URL: <http://www.craigfergusonimages.com/2007/03/80million-lines-of-code-adobe-cs3/>.

38. См. URL: *Moga G. B. Gmail to Use More HTML5 Features*//Google Operating System. June 28, 2010. URL: <http://googlesystem.blogspot.ru/2010/06/gmail-to-use-more-html5-features.html>.

39. См.: Динамический сайт//Wikipedia.org. https://ru.wikipedia.org/wiki/Динамический_сайт.

40. См.: Трехуровневая архитектура//Wikipedia.org. URL: https://ru.wikipedia.org/wiki/Трехуровневая_архитектура.

взаимодействий между десятками или даже сотнями отдельных программных процессов. В 2009 году Google начал разработку нового языка программирования Go, предназначенного для «большого программного обеспечения» — «больших программ, написанных множеством разработчиков и со временем разрастающихся, чтобы поддерживать облачные сетевые сервисы: одним словом, серверное ПО»⁴¹.

Сложность и распределенная архитектура современного крупномасштабного ПО ставят серьезные проблемы перед тезисом о том, что изучать интерактивные медиа — значит изучать код их ПО. Как бы то ни было, даже если речь идет об относительно короткой программе, а критик вполне осознает при чтении кода, что именно должна делать программа, такое понимание логической структуры программы нельзя перевести в какое-либо представление об актуальном опыте пользователя. Если бы это было возможно, то экстенсивное тестирование с участием настоящих пользователей, которое проводят до выпуска нового продукта все компании, производящие ПО и медиа (все что угодно — от нового программного приложения до новой игры), было бы излишне, а проблемного поля взаимодействий человек–компьютер просто не было бы.

Привлекательность «чтения кода» в качестве подхода для гуманитарных наук состоит в том, что он создает иллюзию наличия статичного и конечного текста, который можно изучить, то есть распечатки программы. Но это все же иллюзия, и мы вынуждены принять фундаментальную изменчивость «выполнения программы». Поэтому вместо анализа кода как абстрактной сущности нам следует отслеживать, как он осуществляется или «выполняется» в конкретных пользовательских сессиях. Конечно, это отслеживание может сочетаться с такими записями чувственного измерения интерактивного опыта, как упоминавшаяся мной ранее видеосъемка геймплея на экране. Если говорить языком лингвистики, то вместо того, чтобы думать о коде как языке, нам стоит исследовать его как речь.

Марк Марино и другие ученые, работающие в парадигме «критических исследований кода», разрабатывали более нюансированные, теоретически строгие и богатые идеи о том, что зна-

41. Роб Пайк, цит. по URL: <http://readwrite.com/author/joe-brockmeier-1; Brockmeier J. Permanent link to Google's Go Programming Language Grows Up: Now What? // ReadWrite. March 29, 2012. URL: http://readwrite.com/2012/03/29/googles-go-programming-languag>.

чит «читать код», поэтому данная критика направлена только против наивной версии этой идеи, которая иногда встречается в гуманитарных науках. В своем введении в эту парадигму Марино уточняет, что «код» — это не только распечатка программы, но и программная архитектура⁴², что имеет решающее значение в масштабных программных системах.

Есть другой концептуальный подход — можно сравнить компьютерный код с нотной записью, которая интерпретируется в процессе исполнения (этим предполагается, что музыкальную теорию можно использовать для понимания программной культуры). Хотя такой подход и выглядит многообещающим, он тоже ограничен, ведь не может работать с самым важным измерением опыта программных медиа — интерактивностью.

Ключевой задачей для исследований ПО, несомненно, является развитие методов изучения современного ПО, редуцирующих его сложность к более абстрактным репрезентациям, которые затем можно обсуждать в статьях, на конференциях и в публичных дискуссиях с участием тех, кто программистом не является. Впрочем, я не думаю, что мы сможем решить эту проблему в обозримом будущем, учитывая сложность реальных программных систем и тот факт, что по крайней мере сейчас лишь немногие исследователи медиа и культуры подкованы в программировании. В области разработки ПО уже есть техники, позволяющие представлять алгоритмы и программы, используя высокоуровневые репрезентации вроде псевдокода, блок-схем и *UML*-диаграмм, однако понимание этих представлений требует знания основ программирования.

4.

Хотя проект изучения и интерпретации программного кода — это вызов, мы можем воспользоваться общим свойством медиа, созданных на основе программ и осуществляющихся с их помощью, которое я бы назвал «самоописанием». Другой способ выразить эту идею — сказать, что «тексты» цифровых медиа содержат высказывания, сформулированные как на объектном языке, так и на метаязыке.

Что я имею в виду под «самоописанием»? Практически во всех типах цифрового медиа некоторые аспекты их структуры и динамического поведения, возможный спектр пользовательского

42. Marino M. Critical Code Studies//Electronic Book Review. 12.04.2006. URL: <http://www.electronicbookreview.com/thread/electropoetics/codology>.

опыта взаимодействия и семантика описаны на «программном уровне». Этот уровень может принимать разные формы. *HTML*-код веб-страницы, *XML*-документ, «проектный файл» *Adobe After Effects*, описывающий детали проекта динамического изображения, или действующий программный код, такой как код видеоигры. Заметьте, программный «код» — лишь одна из таких форм, и на самом деле его сложнее анализировать (по только что перечисленным причинам), чем другие типы «программного уровня» — языки разметки (*HTML*, *XHTML*, Вики-разметка) или шаблоны блогов.

Конечно, во многих форматах «старых медиа» тоже есть элементы самоописания, упрощающие автоматический анализ. Например, в тексте пьесы эксплицитно даются имена каждому из персонажей. Статья разделена на части, и если имена этих частей отчетливо обозначены и понятны, то они дают краткое резюме структуры и содержания статьи. Но в цифровых медиа такие элементы самоописания более систематичны, подробны и обязательны. Так, *HTML*-код веб-страницы явным образом задает те типы медиа, что могут содержаться на этой странице. Последующие примеры, взятые со страницы на *software-studies.com*, показывают код для включения изображения и видеофайла:

```
<a href="http://www.flickr.com/photos/culture-
vis/5109394222/" title="Manga Style Space by culture-
vis, on Flickr"></a>
<iframe width="560" height="315" src="http://www.you-
tube.com/embed/hd3HMRAIxw4?rel=0" frameborder="0"
allowfullscreen></iframe>
```

А вот другой, даже более наглядный пример черты самоописания в цифровых медиа — фрагмент из файла на языке *Postscript*, описывающий линию:

```
0.1 setlinewidth
2 2 newpath moveto
3 3 lineto
3 4 lineto
2 4 lineto
0 setgray
stroke
```

Если мы знаем язык *Postscript*, то поймем, что эта часть файла задает несколько соединенных прямых линий, и нам не понадобится видеть саму заданную картинку. Каково основание этой точности и ясности кодированных описаний содержания цифровых медиафайлов? Эти описания предназначены не для человеческого глаза. Что, однако, вовсе не означает, будто мы должны согласиться со знаменитыми словами в начале статьи Фридриха Киттлера «нет никаких программ» (1985):

... кипы написанных текстов, включая этот, больше не существуют в воспринимаемых времени и пространстве — они в транзисторных элементах компьютерной памяти.

Они читаются программой, производящей «медиа» (то, что мы смотрим, слышим, по чему перемещаемся и с чем взаимодействуем). Компьютерным программам нужны эксплицитные инструкции для любых действий — отобразить ли картинку и видео на веб-странице (первый пример) или просто нарисовать векторную линию (второй пример).

Это рассуждение можно использовать для следующего определения цифровых медиа: *медиаобъект состоит из данных и набора формализованных инструкций, которые описывают эти данные и указывают медиаприложениям, как их обрабатывать.*

Акцент на этом фундаментальном свойстве самоописания, характерном для всех типов цифровых медиа, делает их доступными для существующих техник анализа «больших массивов данных». Вместо того чтобы думать об интерпретации конкретных случаев программного кода (что обычно и недоступно для коммерческого ПО), мы можем использовать компьютеры для анализа паттернов использования элементов самоописания больших массивов цифровых медиадокументов. Например, Мэтью Фуллер предложил картографировать эволюцию веб-дизайна, анализируя частоту *HTML*-тегов в большом количестве исторических снимков экрана с веб-страницами, которые доступны на *Archive.org*.

Можно представить себе изучение паттернов архитектурного дизайна на материале раздела историй команд, который поддерживается всеми программами для дизайна. К примеру, можно попросить студентов-архитекторов предоставить сохраненные истории команд, использовавшихся ими при работе над своими проектами, а затем проанализировать паттерны в их использовании. Можно распространить эту идею на все остальные типы

дизайна и авторской работы. Вспомните мечту о Семантическом Вебе — добавить семантическую разметку к каждому документу в сети. И если мы когда-либо хоть отдаленно приблизимся к этому, мы получим еще один удивительный инструмент для анализа культурных паттернов.

Настоящий метод можно применить и к изучению самого компьютерного кода. Вместо анализа распечаток кода вручную мы можем использовать методологию «больших массивов данных», автоматически «прочитывая код» и извлекая паттерны. Так, Джереми Дуглас предложил изучать распространение медиатехник и конвенций, отслеживая использование программных библиотек в разных программных системах и проектах⁴³. Очевидно, что эта блестящая идея работает, только если у нас будет доступ к достаточному количеству программных распечаток. К счастью, благодаря движению за открытые исходные коды в каждой части программного мира мы найдем широко используемые и доступные для такого анализа эквиваленты коммерческих продуктов, например браузер *Firefox* (рыночная доля по разным оценкам от 16 до 21%⁴⁴), программа *Blender 3D* (пятое место по частоте использования в 2010 году⁴⁵). Другие объекты для применения этого метода «большого кода» (по аналогии с «большими массивами данных») — сценарии, написанные на *Perl*, *Python*, *MEL*, *Javascript* и других языках, шаблоны *Blogger* и *WordPress*, а также любые другие ПО, распространяемые в формах, доступных для прочтения человеком (в противоположность бинарным).

Итак, свойство самоописания, характерное для всех цифровых медиа, открывает новые возможности культурного анализа «больших массивов данных». Вместо анализа видимых аспектов интерактивных артефактов (к примеру, графического и интерактивного дизайна веб-сайта, его текста и любых других медиаэлементов) мы можем заняться разбором и визуализацией структур и паттернов «программных уровней» этих артефактов.

43. *Douglass J.* Include Genre. Presentation at «Softwhere 2008» workshop, University of California, San Diego.

44. См.: Mozilla Firefox // Wikipedia.org. URL: https://ru.wikipedia.org/wiki/Mozilla_Firefox.

45. *Community Reporter.* CG Survey Initial Results In // Blendernation.com. April 18, 2010 URL: www./2010/04/18/cg-survey-initial-results-in/.

How to Follow Software Users

Lev Manovich. PhD in Visual and Cultural Studies, Director of the Software Studies Initiative at the University of California, San Diego (UCSD), Professor at The Graduate Center of the City University of New York. Address: 365 Fifth Avenue, New York, NY 10016, USA. E-mail: manovich.lev@gmail.com.

Keywords: software studies; media theory; big data; software; interactivity; open source; critical code studies; cultural patterns.

Software has replaced a diverse array of physical, mechanical, electronic technologies used before the 21st century to create, store, distribute and interact with cultural artifacts. It has become our interface to the world, to others, to our memory and our imagination—a universal language through which the world speaks, and a universal engine that the world runs on. What electricity and the internal combustion engine were to the early 20th century, software is to the early 21st century. This article looks at the cut-

ting edge of modern media research. It discusses the problems of analysing interactive, software-driven processes of media consumption and creation—in other words, the “space” between users and programs. It also asks how the new paradigm of “digital humanities” (quantitative analysis of cultural datasets) can be used to study these interactive processes. Three questions are considered in particular: 1) What is the “data” in interactive media? Software code as it executes, the records of user interactions (for example, clicks and cursor movements), the video recording of a user’s screen, a user’s brain activity as captured by an EEG or fMRI, or something else? 2) Can analysis of software system code give us a detailed understanding of interactive cultural experiences? 3) Who has access to detailed records of user interactions with cultural artifacts and services on the web, and what are the implications of being able to analyze these data?

References

- About Software Studies. *SoftwareStudies.com*, May 2007. Available at: <http://lab.softwarestudies.com/2007/05/about-software-studies-ucsd.html>.
- Brockmeier J. Permanent link to Google’s Go Programming Language Grows Up: Now What? *ReadWrite*, March 29, 2012. Available at: <http://readwrite.com/2012/03/29/googles-go-programming-languag>.
- Community Reporter. CG Survey Initial Results In. *BlenderNation.com*, April 18, 2010. Available at: <http://blender-nation.com/2010/04/18/cg-survey-initial-results-in/>.
- Counter-mapping. *Wikipedia.org*. Available at: <http://en.wikipedia.org/wiki/Counter-mapping>.
- Crawford J. On the Future of Books. *Google Books Search*, October 14, 2010. Available at: <http://booksearch.blogspot.ru/2010/10/on-future-of-books.html>.
- Darwinian Literary Studies. *Wikipedia.org*. Available at: http://en.wikipedia.org/wiki/Darwinian_literary_studies.
- Davidson J., Liebold B., Liu J., Nandy P., Vleet T. V., Gargi U., Gupta S., Yu He, Lambert M., Livingston B., Sampath D. The YouTube Video Recommendation System. *Proceedings of the 4th ACM conference on Recommender Systems (RecSys ’10)*, New York, ACM, 2010, pp. 293–296.
- Dinamicheskii sait [Dynamic Web Page]. *Wikipedia.org*. Available at: http://ru.wikipedia.org/wiki/Динамический_сайт.
- Douglass J. Include Genre. Presentation at “Softwhere 2008” workshop, University of California, San Diego.

- Foys M. K. Digital Keys for Unlocking the Humanities' Riches. *The New York Times*, November 16, 2010. Available at: <http://nytimes.com/2010/11/17/arts/17digital.html>.
- General Perspective Projection. *Wikipedia.org*. Available at: http://en.wikipedia.org/wiki/General_Perspective_projection.
- Google Planeta Zemlia [Google Planet Earth]. *Wikipedia.org*. Available at: http://ru.wikipedia.org/wiki/Google_Планета_Земля.
- Hall S. Encoding/decoding. *Culture, Media, Language* (ed. Centre for Contemporary Cultural Studies), London, Hutchinson, 1980.
- Hasson U., Landesman O., Knappmeyer B., Vallines I., Rubin N., Heeger D. J. Neurocinematics: The Neuroscience of Film. *Projections*, Summer 2008, vol. 2, iss. 1, pp. 1–26. Available at: http://cns.nyu.edu/~nava/MyPubs/Hasson-etal_NeuroCinematics2008.pdf.
- Kessler F., Schaffer M. T. Navigating YouTube: Constituting a Hybrid Information Management System. *The YouTube Reader* (eds J. Burgess, P. G. Lange, M. T. Schafer), Stockholm, National Library of Sweden, 2009.
- Kolichestvo strok koda [Source Lines of Code]. *Wikipedia.org*. Available at: http://ru.wikipedia.org/wiki/Количество_строк_кода.
- Latour B. *Nauka v dejstvii: sleduja za uchenymi i inzhenerami vnutri obshchestva* [Science in Action: How to Follow Scientists and Engineers through Society], Saint Petersburg, European University at St. Petersburg, 2013.
- List of Academic Databases and Search Engines. *Wikipedia.org*. Available at: http://en.wikipedia.org/wiki/List_of_academic_databases_and_search_engines.
- Manovich L. How to Follow Software Users? *SoftwareStudies.com*, March 2012. Available at: http://softwarestudies.com/cultural_analytics/Manovich.How_to_Follow_Software_Users.doc.
- Manovich L. *Software Takes Command*, London, Continuum, 2013.
- Marino M. Critical Code Studies. *Electronic Book Review*, April 12, 2006. Available at: <http://electronic-bookreview.com/thread/electropoetics/codology>.
- Middler B. Generations of Game Analytics, Achievements and High Scores. *Eludamos. Journal for Computer Game Culture*, 2009, vol. 3, no. 2, pp. 177–194.
- Moga G. B. Gmail to Use More HTML5 Features. *Google Operating System*, June 28, 2010. Available at: <http://googlesystem.blogspot.ru/2010/06/gmail-to-use-more-html5-features.html>.
- Mozilla Firefox. *Wikipedia.org*. Available at: http://ru.wikipedia.org/wiki/Mozilla_Firefox.
- Penley C. Feminism, Psychoanalysis and the Study of Popular Culture. *Cultural Studies* (ed. L. Grossberg), London, Routledge, 1992, pp. 479–500.
- Randall K. Rise of Neurocinema: How Hollywood Studios Harness Your Brainwaves to Win Oscars. *FastCompany.com*, February 25, 2011. Available at: <http://fastcompany.com/1731055/rise-neurocinema-how-hollywood-studios-harness-your-brainwaves-win-oscars>.
- Rekomendatel'naia sistema [Recommender System]. *Wikipedia.org*. Available at: http://ru.wikipedia.org/wiki/Рекомендательная_система.
- Semantic Targeting. *Wikipedia.org*. Available at: http://en.wikipedia.org/wiki/Semantic_targeting.
- Shannon C. E. A Mathematical Theory of Communication. *Bell System Technical Journal*, 1948, vol. 27.
- Shannon C. E., Weaver W. *The Mathematical Theory of Communication*, Urbana, IL, University of Illinois Press, 1949.
- Social Search. *Wikipedia.org*. Available at: http://en.wikipedia.org/wiki/Social_search.

- Trekhurovnevaia arkhitektura [Three-Tier Architecture]. *Wikipedia.org*. Available at: https://ru.wikipedia.org/wiki/Трёхуровневая_архитектура.
- Веб 2.0 [Web 2.0]. *Wikipedia.org*. Available at: http://ru.wikipedia.org/wiki/Веб_2.0.
- Zhou R., Khemmarat S., Gao L. The Impact of YouTube Recommendation System on Video Views. *Proceedings of the 10th Annual Conference on Internet Measurement (IMC '10)*, New York, ACM, 2010, pp. 404–410.