

УДК 004.923

ВИЗУАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ СРЕДСТВ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ ГЕОПРОСТРАНСТВЕННЫХ ДАННЫХ. ОБЗОР И ПЕРСПЕКТИВЫ РАЗРАБОТКИ

Авербух Владимир Лазаревич^{1,2},
averbukh@imm.uran.ru

Авербух Наталья Владимировна²,
natalya_averbukh@mail.ru

Васёв Павел Александрович¹,
vasev@imm.uran.ru

Гвоздарев Илья Леонидович²,
ilyagvz@gmail.com

Левчук Георгий Иванович²,
levchukgeorgy1@gmail.com

Мелкозёров Леонид Олегович²,
leoghost@mail.ru

¹ Институт математики и механики им. Н.Н. Красовского УрО РАН,
Россия, 620990, г. Екатеринбург, ул. Софьи Ковалевской, 16.

² Уральский Федеральный Университет им. первого Президента России Б.Н. Ельцина,
Россия, 620002, г. Екатеринбург, ул. Мира, 19.

Актуальность. В условиях глобальной цифровизации, когда становятся актуальными такие проекты, как цифровая земля, компьютерная визуализация является важным элементом компьютерного моделирования как в научных исследованиях, так и в технике, в частности, в областях геоинформатики, природо- и недропользования, исследования георесурсов. Также компьютерная визуализация необходима при представлении «больших данных», например, накопленных в результате геологических исследований больших пространств. Компьютерная визуализация как самостоятельная дисциплина развивается уже более тридцати лет. В настоящее время в компьютерной визуализации активно используются средства виртуальной реальности, которые и оперируют геопространственными данными. Виртуальная реальность может использоваться при представлении данных о Земле. Возможно создание трёхмерных отображений графических объектов, представляющих элементы ландшафта. Одной из подобластей компьютерной визуализации является визуализация программного обеспечения (Software Visualization). Визуализация программного обеспечения необходима при представлении сложных программных комплексов, связанных с проектированием и разработкой в различных областях техники, включая энергоэффективные технологии.

Объекты. Работа посвящена исследованиям и разработкам в области визуализации программного обеспечения на базе виртуальной реальности. Эти исследования могут применяться в целом ряде областей, связанных с геоинформатикой, а также в области исследования георесурсов.

Методы: методы теории компьютерной визуализации, а именно элементы теории метафоры визуализации.

Цель: поиск подходов, позволяющих эффективно использовать виртуальную реальность для решения сложных проблем, встающих перед специалистами в области программного обеспечения, связанного с большими объёмами сложно структурированных данных, в том числе больших геопространственных данных.

Результаты. Проведен обзор развития визуализации программного обеспечения и описаны проекты в данном направлении, разработанные за последние десятилетия. Приводятся разработанные авторами проекты систем визуального программирования и визуализации программного обеспечения на базе виртуальной реальности с использованием метафор космического пространства и расширенной метафоры города. Рассматриваются психологические аспекты человеческого фактора в разработке систем визуализации программного обеспечения с использованием виртуальной реальности, включая использование теории деятельности. Возникает проблема описания деятельности программиста в качестве пользователя систем визуализации программного обеспечения. Разработка программного обеспечения для реальных технических задач, например для нужд инженерной геологии или прогнозирования георесурсов, требует специализации, а то и персонализации визуальных систем. В результате проведенного анализа можно заключить, что мы находимся на первых этапах важных исследований.

Ключевые слова:

Программное обеспечение для задач геоинформатики, визуализация программного обеспечения, визуальное программирование, виртуальная реальность, метафоры визуализации, человеческий фактор, теория деятельности, геопространственные данные.

Введение

Современная геоинформатика становится более актуальной в связи с развитием таких проектов, как цифровая земля. Необходимо развивать ресурсы для визуализации сложных цифровых объектов реального мира. Развитие геоинформационных систем позволило расширить аппарат работы с пространственными данными с одной стороны и стало частью программных решений для оперирования координатно-распределенными данными с другой стороны, позволяя работать с многомерными представлениями географически распределенных объектов, сцен и виртуальных пространств. Именно поэтому важно развивать принципиально новые подходы к построению больших пространственных систем, каковыми являются геоинформационные системы, а новые интерфейсные решения – это качественный скачок в работе с пространственными многопараметрическими структурами, что является актуальным в развитии геоинформационных технологий.

Современные технологии визуализации, включая виртуальную реальность, позволяют расширить возможности геоинформационных систем. Компьютерная визуализация является важным элементом компьютерного моделирования как в научных исследованиях, так и в технике, в частности, в областях природо- и недропользования, исследования георесурсов. Также компьютерная визуализация необходима при представлении «больших данных», например, накопленных в результате геологических исследований больших пространств. Компьютерная визуализация как самостоятельная дисциплина развивается уже более тридцати лет. Одной из подобластей компьютерной визуализации является визуализация программного обеспечения (Software Visualization). Визуализация программного обеспечения необходима при представлении сложных программных комплексов, связанных с проектированием и разработкой в различных областях техники, включая энергоэффективные технологии.

Работа посвящена проблематике визуализации программного обеспечения и современному состоянию этой дисциплины в связи с использованием сред виртуальной реальности, являющихся средами для работы с пространственными данными. Этой теме посвящено большое количество публикаций и выступлений на различных конференциях. Ежегодно проводятся конференции по визуализации программного обеспечения VISSOFT [1]. В системах визуализации программного обеспечения активно используются методики трехмерной графики, включая среды виртуальной реальности. Сегодня такие системы расширяют границы применения геоинформатики и теории работы с пространственно-координированными данными. Географические явления и объекты служат источником новых метафор визуализации программного обеспечения, в частности, метафор города и ландшафта, о которых будет рассказано ниже.

В следующем разделе рассматриваются начальные этапы становления дисциплины. Затем приводятся

некоторые теоретические положения дисциплины «компьютерная визуализация». Даются описания таких метафор визуализации, как метафора вселенной (космоса) в геоцентрическом и гелиоцентрическом вариантах и расширенная метафора города. Далее делается обзор систем визуализации программного обеспечения на базе сред виртуальной реальности, оперирующих геопространственными данными. Анализ этих систем приводит к необходимости проектирования и разработки собственных прототипов, а также проведения исследований возникающих возможностей и проблем.

Становление дисциплины «визуализация программного обеспечения»

Визуализация программного обеспечения является подобластью дисциплины «компьютерная визуализация» и использует компьютерную графику и средства человеко-машинного взаимодействия как для спецификации и представления программных объектов в процессе создания программ (визуальное программирование), так и для лучшего уяснения понятий в процессе отладки и эффективной эксплуатации программного обеспечения (визуализация программ).

Первые системы визуального программирования и визуализации программ, которые можно отнести к визуализации программного обеспечения, появились в конце 1970-х – начале 1980-х гг. Появление этой дисциплины связано с практикой и идеями графического представления программ, основанных на активном использовании программистами блок-схем (графов потоков управления) и графов потоков данных. В это время намечалась перспектива массовой компьютеризации и казалось, что графическое представление программ резко облегчит разработку и сопровождение программного обеспечения. В число систем визуализации программного обеспечения обычно включаются системы визуального программирования, системы визуализации программирования. Выдвигалась идея визуализации, заключающаяся в том, что и разработка программ, и их отладка, и их сопровождение должны осуществляться в рамках единой системы с одинаковым графическим представлением программных сущностей. В нашей стране и за рубежом были разработаны системы анимации алгоритмов визуального программирования и визуализации программных комплексов. При этом использовалась образность, на основе традиционных («бумажных») представлений структур графов, схем и диаграмм различного типа [2–7].

Описание новой дисциплины было осуществлено в начале 1990-х гг. [8–12]. Несмотря на использование примитивной (по нынешним временам) графики в 1980–1990-х гг., были разработаны интересные системы визуального программирования и полноценные визуальные отладчики. В конце 1980-х и начале 1990-х гг. начали бурно развиваться параллельные вычисления, для которых был разработан целый ряд систем визуализации отладки правильности и отладки эффективности, а также полноценных языков визуального программирования. Языки визуального про-

граммирования подразделялись на диаграмматические языки, использующие схемы и диаграммы, и иконические языки, применяющие естественные образы (пиктограммы) для представления объектов и действий в программах. В основе видов отображения, использованных в системах визуализации программирования, как правило, лежали традиционные методики визуализации – рисование диаграмм и графов различного типа, а также графиков, отражающих те или иные параметры работы программ. Иногда использовалась образность, естественная для отлаживаемого приложения, как, например, в отладчике *Visual* для отладки программы, которая моделирует поведение хищных и нехищных рыб, на параллельном вычислительном комплексе. Среда, в которой действуют рыбы, представляется как сетка, состоящая из квадратов с рыбами. Один из процессоров управлял частью сетки из 4*4 квадратов. Использование визуального отладчика позволило наблюдать разные участки среды и получить информацию о работе параллельной программы [13].

К началу 2000-х гг. визуализация программного обеспечения (*Software Visualization*) оформилась как самостоятельная дисциплина. Ей посвящены полноценные монографии, изданные в этот период [14, 15].

В то же время появились публикации, в которых указывалось на кризис в области визуализации программного обеспечения, например [16].

Возможно, это связано с тем, что использование традиционных («бумажных») методов двумерной графики очень быстро стало ограничением для систем визуализации программного обеспечения. Даже сравнительно небольшие по нынешним меркам отображения, например, сотни связанных между собой объектов, были сложны как для восприятия, так и для интерпретации. Для выхода из создавшейся ситуации активно развивались методы, основанные на использовании трехмерной графики [17, 18]. Развивались теоретические подходы к созданию систем визуализации программного обеспечения, в частности теория метафор визуализации. На основе таких метафор строились трехмерные представления программных объектов [19]. Большой популярностью пользовалась метафора города [20]. Были предприняты попытки применения средств виртуальной реальности для визуализации программного обеспечения.

В следующих разделах мы подробнее рассмотрим теоретические аспекты метафоры визуализации и примеры того, как использовать виртуальную реальность для визуализации программного обеспечения. А затем приведем примеры разработанных нами прототипов таких систем на базе ряда геопространственных метафор с возможным использованием сред виртуальной реальности.

Метафоры визуализации

Определим понятие метафоры визуализации. Метафора – это основная идея сближения понятий прикладной области с той или иной образностью. Метафоры визуализации служат основой конкретных ви-

дов отображения соответствующих систем. Именно проектирование видов отображения является важной частью для определения когнитивной составляющей конкретной специализированной системы визуализации. Метафоры во многом служат основой деятельности пользователя программной системы и его восприятия объектов и операций над ними. Среди метафор, используемых в системах визуализации программного обеспечения, большой популярностью пользуются пространственные метафоры, в частности метафора города. Примеры применения метафоры города приведены в работах [20–23]. Пространственные (трехмерные) метафоры, такие как метафора города, ландшафта и т. п., активно применяются при разработке видов отображения в системах визуализации, используемых для мониторинга, тестирования и отладки параллельных и распределенных программ, а также программ обработки событий и обеспечения реакций на них. Оценить возможность применения той или иной метафоры для конкретных приложений можно за счет анализа их свойств. В работе [24] рассматриваются свойства ряда метафор, используемых в системах визуализации программного обеспечения, в частности геопространственных метафоры города и метафоры частицы.

Метафора города и близкая к ней метафора ландшафта обладают такими свойствами, как *неограниченный контекст, организация внутренней структуры, естественность, устойчивость к масштабированию*. Неограниченность контекста означает, что при визуализации большого объема данных можно увидеть всё изображение и легко выделить ключевые места. Использование метафоры города предполагает структуризацию исходных данных за счет внутренней структуры города, содержащего отдельные кварталы, улицы и дома. Естественность метафор влечет как простоту ориентации в пространстве, так и легкость навигации. Масштаб представления информации в рамках метафоры города и ландшафта можно варьировать, в частности, за счёт изменения высоты, на которой располагается точка наблюдения. В системах визуализации программного обеспечения в рамках метафор города и ландшафта транспортные магистрали часто используются для представления потоков управления и данных, а также различных связей между объектами и частями программы. Для определения особенностей программы, в частности ошибок, их можно выделять формой, цветом или размером.

При использовании данных метафор в системах на базе *виртуальной реальности* возможна реализация полета над городом, а также «вход» в отдельные помещения, где представлена необходимая визуальная информация.

Ниже мы рассмотрим возможности использования расширенной метафоры города при разработке конкретной системы визуализации программного обеспечения с возможным использованием виртуальной реальности. Кроме того, будут использоваться две «космические» метафоры, свойства которых будут кратко описаны ниже. Но перед этим рассмотрим

предыдущий опыт использования средств виртуальной реальности для решения различных задач по визуализации программного обеспечения.

Визуализация программного обеспечения на базе сред виртуальной реальности. Обзор существующих решений

В начале 1990-х гг. была разработана интересная система визуализации программного обеспечения Avatar [25], активно использующая средства виртуальной реальности и метафоры комнаты и здания в трехмерном варианте и функционирующая на базе среды виртуальной реальности типа CAVE. Система Avatar была предназначена для отладки производительности параллельных программ и позволяла представлять большие объемы данных о производительности параллельных процессов, полученных в ходе работы суперкомпьютера. В системе используется понятие «scattercube», которое является трехмерным обобщением традиционного вида отображения «scatterplot» – точечной диаграммы или диаграммы рассеивания. По ходу работы пользователь как бы оказывается внутри трехмерного помещения, на стены которого проецируется видеоизображение. На пол и стены этого помещения выводятся кривые, описывающие метрики производительности параллельных программ в виде двумерных графиков. Реализовано объединение отдельных элементов в виде scattercube-матрицы, что походит на стеклянный небоскреб, при этом каждая из его комнат содержит графический вывод, характеризующий различные аспекты поведения описываемой параллельной программы. Предусматривался режим прозрачности потолка и пола, что позволяло увидеть соседние данные в соседних «помещениях». Было определено визуальное отображение – «лента истории», для представления последовательности работы процессоров суперкомпьютера. «Виртуальный полет» по небоскребу на основе этой ленты давал возможность исследования совокупных данных о производительности прикладной параллельной программы. Обратим внимание на то, что, несмотря на очень интересную реализацию системы Avatar, данных о ее использовании или продолжении разработок найти не удалось. Это, возможно, связано как с недостаточной информативностью использованных при визуализации данных о производительности процессоров двумерных графиков, так и с возможными неприятными и даже болезненными ощущениями, которые могут быть вызваны у пользователя при «виртуальном полете» по scattercube, который проходит согласно «ленте истории» работы многопроцессорной системы. Эти проявления *киберболезни* (cybersickness – расстройства здоровья, сходного с морской болезнью) часто связываются с отсутствием возможности активного управления событиями виртуальной реальности [26].

Новый интерес к использованию виртуальной реальности в системах визуализации программного обеспечения проявился в самом начале 2000-х гг. В частности, в работах [27, 28] можно найти целый ряд интересных идей в этом направлении. Примерно

десять лет назад начали появляться работы с описанием возможностей использования виртуальной реальности для представления программных объектов на базе метафор города и ландшафта [29, 30]. Уже в этом десятилетии работы по данному направлению резко активизировались.

В работе [31] был описан прототип системы визуализации программного обеспечения на основе метафоры города. В системе использовался шлем виртуальной реальности. Интерфейс был разработан на основе языка жестов. Для первичной оценки системы авторы провели структурированные интервью, где участники должны были решить три задачи на понимание программы и оценить удобство использования жестов, а также указать свои общие впечатления от использования средств виртуальной реальности для понимания смысла программы. Та же исследовательская команда опубликовала работу по использованию физического моделирования на базе 3D-принтинга для задач визуализации программного обеспечения [32]. Визуализация программных комплексов проводится на основе метафоры города. В результате пользователи получают физические модели объектов программы, что повышает возможности их представления и понимания результата работы. Наличие физических моделей также способствует лучшему взаимопониманию участников разработки. Предполагается возможность использования этих моделей в процессе образования.

Метафора города является весьма популярной в системах визуализации программного обеспечения, использующих средства виртуальной реальности. Отметим в этой связи две системы, которые носят весьма схожие имена, – VR City [33] и CityVR [34].

В системе VR City используется модифицированная метафора города для представления программных систем и связанных с ними аналитических данных. Визуализируются статические (метрики) и динамические (трассы) аспекты программ. Пользователи могут наблюдать и взаимодействовать с объектами «города» в среде иммерсивной виртуальной реальности. Предоставляется функция просмотра исходного кода.

В публикации [34], кроме метафоры города, используемой для визуализации программного обеспечения на базе виртуальной реальности, рассматривается *геймификация* (gamification) разработки программного обеспечения. Этот подход подразумевает создание инструментов, которые предоставляют разработчикам программного обеспечения интерфейс, аналогичный интерфейсу компьютерных игр. Проводился анализ того, как разработчики взаимодействуют с системой визуализации CityVR. Разработчики были взволнованы, они почувствовали интерес к работе, определенный вызов, испытали погружение в виртуальный мир при сохранении управляемости системы. Они затратили значительное время при взаимодействии для навигации в виртуальном мире и при выборе нужных элементов программы. Пользователи поняли, что время прошло быстрее, чем в реальности, и поэтому были готовы потратить больше времени, используя программные средства для решения проблемы обеспечения понимания задачи.

Также на *геймификацию* при разработке систем визуализации программного обеспечения на базе виртуальной реальности указывается в статье [35]. Описана среда на базе виртуальной реальности, которая должна обеспечить работу со структурами программного кода с использованием нескольких метафор (включая метафору города) для визуализации, навигации и передачи информации о программном коде в интерактивном режиме. Были реализованы игры, продемонстрировавшие потенциал геймификации для улучшения понимания структурных зависимостей и модуляризации кода. Отметим также публикации этой команды исследователей, посвященные разработке систем визуализации программного обеспечения на базе виртуальной реальности [35, 36] (включая работу по виртуальному программированию с использованием виртуальной реальности [37]). В работе [38] рассматриваются перспективы использования погружения (*immersion*) разработчиков в структуры программ, а в [36] виртуальный полет над программными объектами в рамках различных метафор (в том числе метафор города, ландшафта и космического пространства).

Еще одна работа [39] посвящена визуализации программных систем на базе виртуальной реальности. Инструмент позволяет программистам, менеджерам проектов или заказчикам исследовать архитектуру и получить первое впечатление о размерах компонентов и их зависимостях. В виртуальных сценах используется простая графика – кубы различных размеров и цветов для представления программных объектов с показом зависимостей в виде соединительных линий.

В работе [40] рассматривается использование смешанной реальности для отладки кода. Предлагается интерактивная 3D визуализация потока сообщений, сочетающая традиционные методики отображения с дополненной реальностью на базе шлема виртуальной реальности.

Работа [41] посвящена проблемам оценки юзабилити (пригодности) систем визуализации программного обеспечения. Однако в какой-то мере ее можно представить как развитие проекта CityVR. Система CityAR создана на основе среды иммерсивной расширенной реальности. В рамках метафоры города здания используются для представления классов, а их размеры и цвет кодируют метрики программного обеспечения.

В работе [37] представлена система на базе виртуальной реальности для программистов. Эта система должна обеспечить визуализацию структуры программного кода на базе нескольких метафор. Реализован просмотр текста с использованием реальных клавиатуры и мыши, а также взаимодействие в рамках смешанной реальности. Это обеспечивает поддержку основных задач программирования без выхода из среды виртуальной реальности. Реализован прототип системы. Проведено исследование, демонстрирующее осуществимость идей. Получены результаты эмпирической оценки, демонстрирующие потенциал системы.

В работе [42] описан инструмент визуального программирования с использованием иммерсивной виртуальной реальности. Предполагается использование данной системы для создания встроенных цифровых комплексов. Система позволяет описывать сложные объекты и их связи, задавать логические конструкции, визуализировать в реальном масштабе времени потоки данных между реальными объектами разрабатываемых комплексов. Используется естественная для приложений образность. Первичный анализ экспериментальных результатов показал, что участники различных уровней квалификации могут успешно создавать и отлаживать программы в рамках заданных сценариев.

Отметим также работу, посвященную оценке эффективности систем трехмерной визуализации программного обеспечения [43]. Для эксперимента была реализована трехмерная визуализация на базе метафоры города на стандартном экране компьютера, в среде виртуальной реальности и физической модели с использованием 3D-печати. Участники (которые были разделены на три группы для каждого носителя) провели визуализацию программных систем различных размеров, решили набор задач на понимание результатов и заполнили анкету. Эффективность визуализации была оценена с точки зрения производительности, набора впечатлений и пользовательского опыта. Хотя разработчики, использовавшие физическую визуализацию, затратили меньше времени для идентификации резко отклоняющихся значений, наименьшие сложности были достигнуты при визуализации систем на основе стандартных экранов. При этом разработчики, использовавшие средства виртуальной реальности, получили самые большие впечатления.

Даже из краткого обзора видно, что проблематика использования средств виртуальной реальности для визуализации программного обеспечения вызывает интерес. В то же время их уровень – это уровень разработки прототипов и оценки возможностей эффективности виртуальной реальности для визуализации программного обеспечения. Представляется, что надо углублять исследования и опытные разработки, используя различные метафоры визуализации. Оценку эффективности надо вести с использованием психологических методик, в частности, опирающихся на теорию деятельности.

Разработка прототипов систем визуализации программного обеспечения на базе сред виртуальной реальности

Анализ примеров систем визуализации программного обеспечения на базе сред виртуальной реальности показывает, что, несмотря на 25 лет истории, остается еще много вопросов по применимости этих сред на практике. Все это требует дополнительных исследований и разработок систем с учетом конкретных задач программного обеспечения. В этом разделе описываются прототипы систем визуализации программного обеспечения на базе нескольких пространственных метафор, предполагающих использование сред виртуальной реальности.

Визуализация программного обеспечения на основе метафоры города

Разрабатывается проект визуализации программного обеспечения для нужд разработчиков и тестировщиков с возможным использованием виртуальной реальности. Предполагается использование двух форм представления – графового и представления на основе метафоры города.

Рассмотрим прототип системы визуализации программного обеспечения, которая базируется на расширенной метафоре города. Предполагается расширение метафоры города за счет добавления активных агентов путем ввода параметров в определенные функции и методы. Агенты могут передвигаться внутри города, тем самым определяя, где используются функции, где изменяются и как проходит процесс работы алгоритмов. Таким образом, расширенная метафора создает такие дополнительные свойства, как возможность наблюдения за программными объектами изнутри зданий или помещений, отображающих те или иные программные сущности в ходе перемещения по городу активных агентов.

С помощью агентов можно проводить процессы отладки, тестирования и сравнительного анализа работы кода с различными типами данных, логикой и другими характеристиками, отвечающими за качество кода. А благодаря связям можно отслеживать взаимодействие различных кусков кода между собой и определять тип связи. Появляется возможность пользователю продумывать логику работы этого кода для грамотного использования быстрых или наиболее близких к быстрым связей, которые повышают уровень быстрой реакции алгоритма.

В системе обеспечивается интерактивное взаимодействие с пользователем посредством формирования визуализации выбранного проекта с выведением информации обо всех составляющих единицах кода. Представляется, что для тестировщиков удобнее визуализация на базе метафоры города. Тогда как разработчик, уяснив проблему, может использовать соответствующую визуализацию кода на базе графовых представлений. Поэтому графовое представление используется для исследования кода и получения информации о его структуре на данный момент, а использование метафоры города позволяет получить информацию для коррекции кода. Рассматривается возможность обучения системы за счет сравнения кусков кода, поиска примерно одинаковых мест. Это позволит предсказывать влияние изменений кода на его работу.

При разработке программного обеспечения, контролирующего работу реальных изделий и механизмов, целесообразно визуализировать их работу, используя естественную графику. Виртуальная реальность позволяет построить отображения изнутри объектов, размещать в трехмерной модели различные датчики и увидеть, как функционирует устройство под управлением программы.

Разработан прототип системы визуализации программного обеспечения. Его интерфейс представляет собой меню с возможностью выбора типа деятельно-

сти программиста (разработчик, тестировщик и т. п.) с уровнями доступа для разных видов работ для решения различных задач. После выбора типа деятельности пользователь может выбрать либо проект, над которым хочет поработать, либо проект, который хочет исследовать, а может начать писать новый проект.

Представляется, что при создании нового проекта можно использовать и другие метафоры, например, одна из «космических» метафор (описанная ниже гелиоцентрическая метафора) может позволить визуально корректировать структуру и логику алгоритмов в коде для быстрой реализации начальной версии программного проекта.

Если выбран уже существующий проект, то предоставляется возможность работы в рамках виртуальной реальности с трехмерной визуализацией структуры кода на базе метафоры города, что дает краткое и качественное описание логики работы алгоритмов данного проекта. При выборе здания, представляющего часть описания логики алгоритма, высвечивается соответствующее визуальное представление с окном самого алгоритма, его описанием, историей изменений и перезапусков версий алгоритма, а также различной статистикой и др. При этом каждый вид деятельности предполагает свое представление информации.

В ходе геопространственной реализации визуального мира в среде виртуальной реальности создается соответствующая сцена, отображающая город, в виде которого и представляется проект, с которым планирует работать программист (рис. 1).

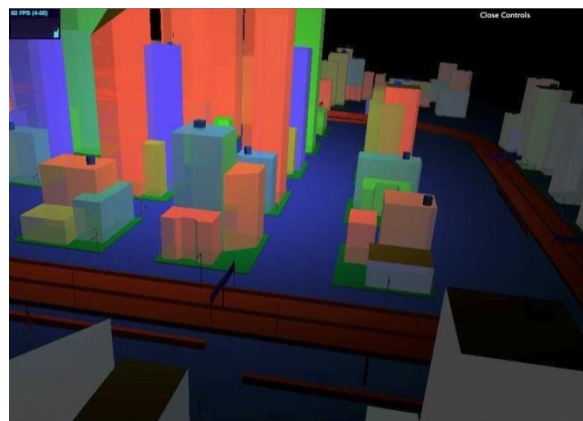


Рис. 1. Представление программного проекта в виде города в виртуальной реальности

Fig. 1. Presentation of a program project as a city in virtual reality

Структура виртуального города соответствует структуре дерева каталогов и файлов разрабатываемого проекта: квартал и подквартал – это папка и подпапка, а здание – файл. В этом случае дороги представляют границы первого уровня подпапок проекта. Следующий уровень программного проекта будет представляться интерьером зданий. Другие объекты программного комплекса могут быть представлены в виде деревьев, водопровода, электростанций и т. п.

В будущем планируется проведение исследований возможностей визуального представления структуры кода в виртуальной реальности с повышением качества изображения и добавления способности визуализировать код различных языков программирования под различными платформами, средами для определения лучшего варианта взаимодействия.

Система визуального программирования на базе метафоры космического пространства

Рассматривается проект системы программирования на базе визуального языка, построенного на принципах объектно-ориентированного программирования. Целью проекта является создание среды визуального программирования с возможностью абстрагироваться от правил и спецификаций конкретного языка, что дает возможность сосредоточиться на решаемой проблеме при написании программы. Таким образом, программист, манипулируя графическими объектами на разных уровнях абстракции, сможет лучше понимать структуру программы и минимизировать число связанных ошибок, таких как несоответствие типов, нарушение синтаксиса и т. п.

В качестве идеи среды визуального программирования выбрана метафора космического пространства в современном понимании с использованием геллиоцентрической картины мира. При этом часть сущностей программы представляется в виде планет, их спутников, колец (типа кольца Сатурна), а также других элементов космического пространства (рис. 2).



Рис. 2. Общий вид среды визуального программирования
Fig. 2. View of the visual programming environment

Среди свойств данного варианта космической метафоры можно отметить:

- контекст, ограниченный только возможностями пространственного восприятия пользователей;
- свободу выбора размещения программных сущностей, при этом появляется возможность двумерного и трехмерного представления в разных ситуациях;
- естественность метафоры, что уменьшает усилия, требуемые для интерпретации результирующего изображения;
- организованность внутренней структуры: структура программного обеспечения (пакеты, классы и их поля и методы, отношения между ними, аргу-

менты и т. д.) может быть спроецирована на структуру вселенной (галактика, звездная система, планета, спутники и кольца);

- масштабируемость.

Данную метафору также можно использовать при разработке систем визуализации программного обеспечения на базе геопространственных сред виртуальной реальности. Виртуальная реальность даст преимущества в навигации, перемещении по структуре классов (космическому пространству) и взаимодействии. Стереоскопичность средств виртуальной реальности позволяет оценить относительный размер объектов и дистанции между ними, что поможет устранить неоднозначность представления сложных программных объектов [28]. Предполагается осуществлять взаимодействие с системой с помощью применяемых в средах виртуальной реальности устройств, а традиционная клавиатура используется для вспомогательных задач, например, для написания идентификаторов. Ставится задача реализации масштабируемости (учитывая разные уровни абстракции), обеспечения навигации по коду и легкого перехода между уровнями абстракции. Увеличение сложности программного обеспечения не должно сильно усложнять графическое представление.

Системы визуализации, использующие этот вариант космической метафоры, должны удовлетворять критериям мантры Шнейдермана, то есть позволять проведение анализа качества визуализации, включающего последовательность повторений операций по обзору, масштабированию, фильтрации и детализации [44], а также операции по выводу зависимостей, получению истории поиска и извлечению подмножества из рассматриваемых данных [45].

В данной системе визуального программирования все пользовательские классы представляются планетами. У каждой планеты (класса) есть два вида отображения: свободный (привычный вид планеты со спутниками и кольцами) и активный (когда класс выбран, выглядит как развертка планеты). Активный вид представляет собой круг с сектором колец слева и упорядоченными спутниками справа. Сектор колец – методы класса. Каждое кольцо – отдельный метод. Внешние кольца – public-методы, внутренние – private и protected. Спутники – поля класса. Сверху вниз сначала идут спутники, более удаленные от планеты – public-поля, менее удаленные – private и protected. Внутри планеты также есть сектор колец, упорядоченные спутники (static-методы и поля), принадлежащие самому классу, и в центре ядро, хранящее все конструкторы класса. Алгоритм (набор инструкций) может быть реализован в виде контекста, то есть в виде области пространства с последовательностью выражений, которые имеют свое визуальное представление. Каждое визуальное представление можно поместить в любом месте этой области. Порядок выполнения предполагается задавать вручную.

В дальнейшем возможно расширение среды визуальной разработки программного обеспечения, например, за счет введения графических аналогов

синтаксических конструкций из языка Java, так, чтобы можно было создавать программы любой сложности. Также можно внедрить поддержку подключения внешних библиотек и классов, созданных вне рамок системы.

Визуализация программного обеспечения на базе геоцентрической метафоры

Для представления иерархически распределенных данных можно использовать другой вариант метафоры космического пространства, опираясь на старинную геоцентрическую модель вселенной. В этой модели постулировалась неподвижная Земля в качестве центра вселенной. Вокруг Земли обращаются прозрачные твердые сферы с прикрепленными к ним небесными телами: сначала Луна, затем Солнце, затем планеты Солнечной системы (каждая планета на своей сфере), а потом звезды. Вся модель содержала набор вложенных друг в друга сфер.

Данная метафора позволяет представлять большие объемы размещенной на многих уровнях информации. В случае использования виртуальной реальности данная метафора обеспечивает хорошую навигацию и перемещение в рамках виртуального мира. Существовало двумерное и трехмерное представление геоцентрической модели, и для задач визуализации программных объектов можно использовать оба этих варианта.

Двумерный вариант был предложен еще в первой половине 1990-х гг. для отображения транспьютерных сетей [46].

Представляется, что для случая использования виртуальной реальности может оказаться удобнее использовать наиболее архаичный трехмерный вариант геоцентрической модели, когда Земля представляется плоской, а небесные тела размещены на полусферах, накрывающих Землю.

В рамках геоцентрической метафоры предложено два прототипа систем визуализации, в двумерном и трехмерном вариантах.

Трехмерный вариант геоцентрической метафоры

Метафора геоцентрической системы в трехмерном пространстве используется для представления данных о функционировании суперкомпьютера в некоторый момент времени. Параллельный суперкомпьютер состоит из набора процессоров. Системный администратор изучает данные о работе суперкомпьютера с целью повысить его эффективность. Визуальное представление в рамках данной метафоры состоит из слов:

0. В центре геоцентрической системы – файловая подсистема суперкомпьютера.
 1. На первой сфере вокруг центра показаны пользователи, задачи которых выполняются в заданный момент на суперкомпьютере.
 2. На второй сфере – выполняющиеся задачи. Размер объекта задачи означает количество вычислительных узлов суперкомпьютера, выделенных задаче.
- Соединения между слоями означают количество операций чтения и записи, которое производят задачи

пользователей по отношению к файловой подсистеме. Чем толще соединение, тем больше таких операций производится. Таким образом, исследователь может наблюдать перечень пользователей и их задач, которые формируют нагрузку файловой подсистемы.

Пример работы описанного выше вида отображения показан на рис. 3. На рисунке видно, что основную нагрузку на файловую систему (зеленая сфера в центре) несут лишь некоторые пользователи. Также видна фактура использования суперкомпьютера – у одних пользователей работает по одной задаче, у других – две–три. Но есть и исключения: пользователь в правой части рисунка запустил порядка 30 мелких задач.

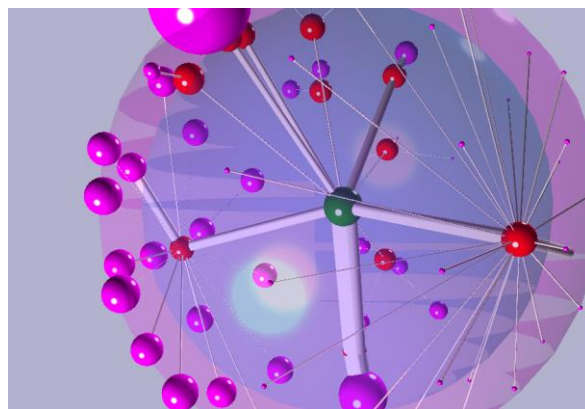


Рис. 3. Визуализации нагрузки на файловую систему суперкомпьютера с использованием геоцентрической метафоры

Fig. 3. Visualization of the load on the file system of a supercomputer using geocentric metaphor

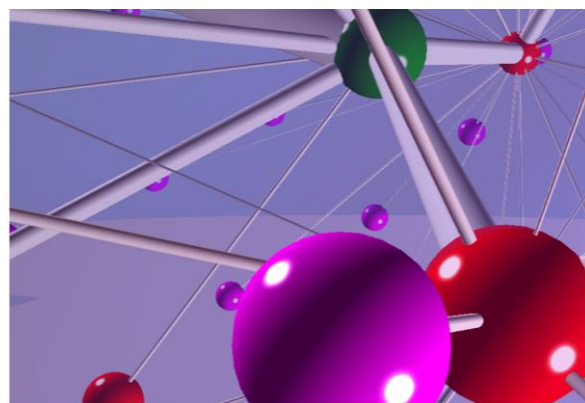


Рис. 4. Фрагмент представления состояния суперкомпьютера в некоторый момент времени, сделанный изнутри трехмерной геоцентрической модели

Fig. 4. Fragment of the supercomputer state at some point in time, made from within a three-dimensional geocentric model

Задачи пользователей, в свою очередь, выполняются на вычислительных узлах суперкомпьютера. Таким образом, можно построить и третью сферу, на которой отображаются вычислительные узлы суперкомпьютера. В этом случае задачи будут показывать-

ся «планетами» единого размера, а вокруг них на третьей сфере – используемые ими узлы. За счет цвета и размера «небесных тел» можно отобразить дополнительные данные о вычислительных процессах.

Программа визуализации в рамках метафоры трехмерной геоцентрической системы построена на базе веб-библиотеки трехмерной графики Viewlang.ru. Данная библиотека опирается на стандарт WebGL. Также в ней обеспечена поддержка технологий виртуальной реальности WebVR. При наличии совместимого устройства (например, Oculus Rift) пользователю программы визуализации доступна кнопка «перейти в режим виртуальной реальности». При ее нажатии графика начинает транслироваться в устройство виртуальной реальности. Также она реагирует на повороты головы и перемещение пользователя. При этом пользователю доступна и обычная навигация с помощью мыши. Пример изображения, которое может наблюдать пользователь в режиме виртуальной реальности, показан на рис. 4.

Двумерный вариант геоцентрической метафоры

Двумерный вариант геоцентрической метафоры можно использовать для представления программ в рамках парадигмы объектно-ориентированного программирования, которая основана на представлении программы в виде совокупности объектов. При этом каждый из них является экземпляром определенного класса, а классы образуют иерархию наследования. Исходя из предположения, что разработка на языках объектно-ориентированного программирования ведется в class-файлах, точка зрения наблюдателя (Земля) может являться текущим рабочим классом. Система интерфейсов может быть отображена как совокупность спутников текущей планеты. В соответствии с представлением о звездах как далеких и неизменных объектах сторонние зависимости и библиотеки, используемые в проектах, могут быть представлены метафорой звезд или созвездий. Ближайшие небесные тела – планеты – представляют другие классы проекта, их размер и вид зависит от различ-

ных свойств класса, его размера, используемого языка, расширения файла.

Интерпретацией множества прозрачных небесных сфер может служить система каскадного размещения кода по признаку частоты использования или уровню связности с текущим классом (рис. 5).

На входе выбирается модуль на языке питон и используется в качестве отправной точки, «Земли». На основе импортированных модулей строятся «планеты» и «звезды». «Планетами» являются зависимости из того же проекта, а «звездами» – внешние библиотеки, в том числе стандартные. По выбору «планеты» можно ознакомиться с содержанием представляемого модуля. В настоящее время реализована версия, принимающая файлы с количеством зависимостей до 20 (из соображений аккуратности и читаемости). Планируется расширение возможностей системы, в частности, предполагается добавить сортировку «планет» по Coupling (того, насколько много импортированных языковых единиц используется в исходном файле) и, возможно, визуальное представление размеров модулей-планет.

Психологический аспект человеческого фактора

Данный раздел посвящен проблематике человеческого фактора в разработке систем визуализации программного обеспечения с использованием геопространственных средств виртуальной реальности. В этом случае человеческий фактор можно разделить на самые разные аспекты, каждый из которых может влиять на работу пользователя с системой, подчас неожиданно для разработчика.

Обращает на себя внимание проблематика восприятия в средах виртуальной реальности. В первую очередь они зависят от физиологических особенностей. Например, люди по-разному воспринимают основу технологии виртуальной реальности – стереоэффект. Некоторые люди вовсе не видят стереоэффекта и не способны, таким образом, использовать преимущества виртуальной реальности, а их способность взаимодействовать с ней весьма ограничена.

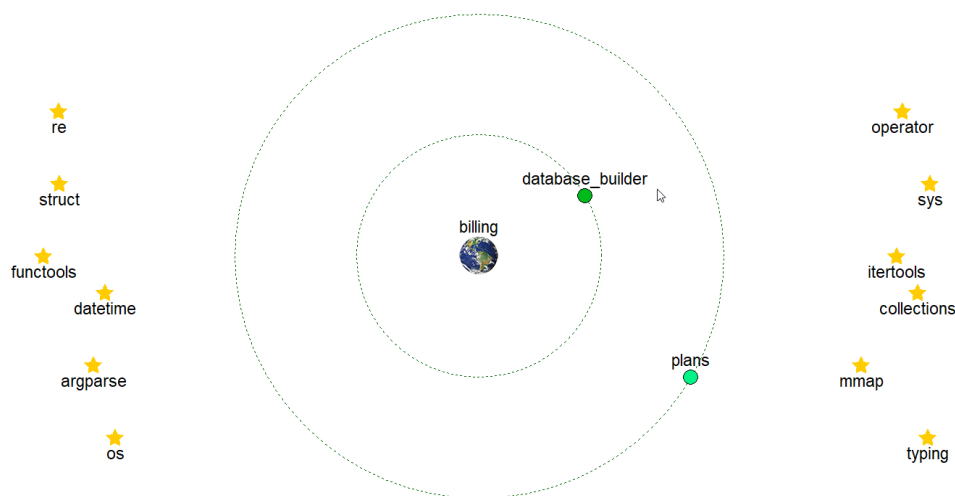


Рис. 5. Двумерное «геоцентрическое» представление программы в рамках объектно-ориентированной парадигмы
Fig. 5. Two-dimensional «geocentric» representation of the program within the object-oriented paradigm

Другой пример, также связанный с физиологией, – это так называемая киберболезнь (cybersickness) – недомогание, которое может появиться при взаимодействии с системами виртуальной реальности. Оно зависит от многих факторов, как системных, так и индивидуальных, а выражается в симптомах, аналогичных морской болезни. Человек, переживающий киберболезнь, очевидно, не способен продолжать работу, а если пытается преодолеть это недомогание, может неважно себя чувствовать в течение нескольких дней после сеанса взаимодействия с виртуальной реальностью. Чтобы предотвратить появление киберболезни, разработчик должен учитывать возможные причины ее появления. Рекомендуются давать пользователю возможность самому инициировать свое движение и оставаться активным во время взаимодействия с виртуальной реальностью. Таким образом, система, в которой пользователь остается на одном месте, а тем или иным образом представленные данные передвигаются вокруг него, если и допустима, то в очень ограниченном режиме. Обзор работ на эту тему дан в [26]. Отметим, что в одной публикации, посвященной разработке систем визуализации программного обеспечения на базе виртуальной реальности [38], указано, что до 30 % участников исследований почувствовали проявление киберболезни.

Вообще проблематика восприятия и других когнитивных процессов чрезвычайно важна для визуализации программного обеспечения, где пользователь решает различные задачи и имеет дело с абстрактной образностью. Эти вопросы подняты в работе [47].

Не менее важными являются вопросы, связанные с интерфейсами сред виртуальной реальности. Очевидно, что для взаимодействия с виртуальной реальностью не всегда будут подходить интерфейсы, рассчитанные для взаимодействия с информацией, выводимой на плоский экран. Интерфейсы, учитывающие работу пользователя в объемной среде, должны учитывать особенности пользователей. Иногда возникают курьезные ситуации, связанные с системами, в которых используется интерфейс на базе захвата движений. Так, например, в некоторых системах, считывающих положение пользователя по положению его головы (в том числе относительно вертикальной оси), низкорослые пользователи могут быть «восприняты» датчиками как намеренно присевшие и им может быть предложена информация, с которой они в данный момент не намеревались взаимодействовать (пример: дно виртуального автомобиля при его обследовании). Опыт показывает, что возникают проблемы в ряде систем отслеживания положения руки, которые рассчитаны на пользователя со стандартной мужской ладонью. Кисть пользователя с меньшим размером руки «воспринимается» такой системой как сжатый кулак, вследствие чего становятся недоступны возможности меню, попасть в которое можно, раскрыв ладонь и растопырив пальцы. Еще один пример представляет собой система, учитывающая положение ног пользователя и «считывающая» длинную юбку как одну ногу. Эти системы нуждаются в

настройке и калибровке, а также в том, чтобы разработчик закладывал широкие возможности в систему.

В общем, из практики следует, что необходимо уделять большое внимание эргономике визуализации и интерфейсов в виртуальной реальности.

Весьма интересен психологический аспект человеческого фактора. Каждый пользователь разрабатываемой системы является носителем определенных психологических характеристик, так или иначе влияющих на его повседневную и профессиональную деятельность. В работах, посвященных разработке визуализации программного обеспечения с использованием виртуальной реальности (например, в [31]), изучается эмоциональный аспект взаимодействия с разработанной системой, эмоциональное состояние пользователей во время взаимодействия с системами визуализации программного обеспечения на базе сред виртуальной и расширенной реальности. Также рассматриваются проблемы юзабилити (пригодности) таких систем [41].

Вместе с изучением эмоционального состояния пользователей следует обратить внимание на особенности в работе различных категорий специалистов в области программного обеспечения. В отечественной психологии давно развивается деятельностная парадигма [48, 49]. Теория деятельности как нельзя лучше подходит для описания влияния человеческого фактора на работу программиста. Ее суть как раз заключается в том, чтобы изучать особенности психики не сами по себе, а применительно к деятельности, то есть целенаправленной активности, в первую очередь трудовой. Деятельностная теория предполагает, что деятельность и личность человека взаимосвязаны и влияют друг на друга: личность формируется под влиянием деятельности и проявляется в деятельности, влияя на процесс и результат.

Использование теории деятельности в процессе разработки программного обеспечения достаточно широко отражено в литературе [50, 51].

На деятельность воздействуют такие психологические характеристики, как поставленная перед собой цель, то есть тот результат, к которому человек стремится, задачи, решение которых ведет к исполнению цели, мотивация, то есть, чего хочет человек, например, познавательная мотивация или финансовая. В отличие от цели, мотиваций может быть несколько, и для эффективности деятельности важно, чтобы они не противоречили друг другу. Также на деятельность воздействуют индивидуальные качества человека, свойства его нервной системы, черты личности и т. п. Важными являются условия протекания деятельности.

На примере упоминаемой выше киберболезни можно показать, как на деятельность влияют создаваемые условия и индивидуальные характеристики. Особенности технологии, более или менее вызывающие киберболезнь, время взаимодействия с виртуальной реальностью, возможность или невозможность самостоятельных действий – все это условия протекания деятельности. Поле зрения пользователя, пол, возраст, подверженность морской болезни (так назы-

ваемая «укачиваемость») – это индивидуальные характеристики пользователя.

Деятельность структурно разбивается на действия и операции.

«Соотношение цели с условиями определяет задачу, которая должна быть определена действием. Целенаправленное человеческое действие является по существу своему решением задачи. Отношение к этим условиям, сочетаясь с отношением к цели, составляет внутреннее психологическое содержание действия» [49. С. 443]. Действия, в свою очередь, распадаются на отдельные операции, которые можно охарактеризовать как связанные друг с другом акты, последовательное выполнение которых ведет к решению поставленной задачи [49].

Преимущество деятельности концепции для описания работы программиста связано с тем, что деятельность ряда категорий программистов, в отличие от деятельности, скажем, инженера или преподавателя, в большей степени поддается разбиению на отдельные действия и операции, являясь, таким образом, хорошо структурно организованной. С другой стороны, не секрет, что взаимодействие с интерфейсами самых разнообразных назначений ставит перед пользователями любых категорий множество задач, не ведущих напрямую к выполнению цели, например, таких, как поиск нужных средств управления или необходимых визуальных объектов. Ориентация на деятельностную парадигму при разработке программной системы могла бы помочь разработчику создать такие условия, чтобы стоящие перед пользователем (в данном случае, поскольку мы говорим о разработке систем визуализации программного обеспечения, это программист той или иной специализации) задачи были связаны только с непосредственными целями той или иной области программирования. Для этого необходимо достаточно четкое описание деятельности различных специалистов (например, проектировщиков, разработчиков, кодировщиков и т. п.). То есть необходимо описание того, какие перед специалистами ставятся цели и задачи в рамках их направления, какие действия и операции они выполняют. Было бы интересно исследование также и мотивации, профессионально-важных качеств (то есть, например, в каком направлении требуется аккуратность, в каком критичность, креативность и т. п.) и других психологических характеристик.

Не вся работа пользователя системы визуализации программного обеспечения является наблюдаемой. Значительная часть работы происходит в уме пользователя и, по сути, взаимодействие с системой визуализации необходимо или для того, чтобы предоставить материал для умственной деятельности или ви-

зуально представить соответствующий результат. Описание внутренней деятельности специалистов в области разработки программных комплексов является важной целью будущих исследований.

Заключение

Разработка прототипов систем на базе виртуальной реальности, реализующих геоинформационный подход, – лишь первый этап исследований. Необходим поиск критериев выбора метафор визуализации и подходов к проектированию видов отображения. Такая работа может вестись на основе исследований в области теории компьютерной визуализации. Отметим, что такие исследования активно ведутся в нашей стране, например, [52–56].

Развитие прикладных геоинформационных систем и технологий невозможно без новых подходов к проектированию, поскольку сложность и объемы пространственных данных катастрофически растут. Возрастает мерность цифровых пространственно-координированных моделей. Использование ассоциативных, понятных пользователю метафор (например, космос, город и т. п.) позволяет оперативно создавать виртуальные ландшафты, сцены, цифровые модели и т. п. Предложенные в работе решения предлагают новый подход к проектированию систем, работающих с пространственными, географическими данными, снижая сложность системы, преодолевая неопределенность и снижая ресурсоемкость процесса проектирования программных, геоинформационных систем и технологий.

В то же время анализ материалов показывает, что остается достаточно много нерешенных проблем. Не ясно, насколько эффективны среды виртуальной реальности для использования в тех или иных задачах визуализации программного обеспечения. Остается множество вопросов к адаптации таких систем с учетом особенностей пользователей, включая индивидуальные ограничения по восприятию и длительному нахождению в средах виртуальной реальности. Идея геймификации, конечно, весьма интересна, но не всегда может быть использована в серьезных разработках. Отсюда вытекает серьезная проблема описания деятельности программиста в качестве пользователя систем визуализации программного обеспечения. Разработка программного обеспечения для реальных технических задач, например, для геоинформационных систем, инженерной геологии или прогнозирования георесурсов, требует специализации, а то и персонализации визуальных систем.

Работа выполнена при финансовой поддержке постановления № 211 Правительства Российской Федерации, контракт № 02.A03.21.0006.

СПИСОК ЛИТЕРАТУРЫ

1. Working Conference on Software Visualization Charter. URL: <http://vissoft-conferences.dcc.uchile.cl/> (дата обращения 15.11.2019).
2. Shu N.C. Visual programming languages: a perspective and dimension analysis // *Visual Languages*. – New York: Plenum Publishing Corporation, 1986. – P. 11–34.
3. Shu N.C. Visual programming. – New York, Van Nostrand Reinhold Company, 1988. – 314 p.
4. Shu N.C. Visual programming: perspectives and approaches // *IBM System Journal*. – 1989. – V. 22. – № 4. – P. 525–547.
5. Brown M. Algorithm animation. – Cambridge, Massachusetts: The MIT Press, 1988. – 200 p.

6. Ковалев А.А., Лизенко С.Л. Графический интерфейс представления алгоритмов и программ // *Управляющие системы и машины*. – 1988. – № 4. – С. 41–47.
7. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. Алгоритмы для правильного мышления. Основы алгоритмизации. – М.: ДМК Пресс, 2012. – 520 p.
8. Price V.A., Small I.S., Baecker R.M. A taxonomy of software visualization // *Proc. of the 25th Hawaii International Conference on System Sciences (HICSS)*. – Hawaii, USA, 1992. – V. 2. – P. 597–606.
9. Price V.A., Baecker R.M., Small I.S. A principled taxonomy of software visualization // *Journal of Visual Languages and Computing*. – 1993. – V. 4. – № 3. – P. 211–266.
10. Ball Th., Eick S.G. Software visualization in the Large // *IEEE Computer*. April 1996. – V. 29. – № 4. – P. 33–43.
11. Авербух В.Л. Визуальное программирование и визуализация программирования. Определения и классификация // *Компьютерная графика в науке и искусстве. Графикон-92: Вторая международная конференция*. – М., 28 сентября – 2 октября 1992. – С. 48–50.
12. Авербух В.Л. Визуализация программного обеспечения. – Екатеринбург. Математико-механический факультет. Уральский Государственный Университет, 1995. – 168 p.
13. Socha D., Bailey M.L., Notkin D. Voyeur: graphical views of parallel programs // *Proc. of the ACM SIGPLAN and SIGOPS Workshop on Parallel and Distributed Debugging*. – Madison, Wisconsin, May 5–6 1988. – V. 24. – № 1. – P. 206–215.
14. Software visualization. From theory to practice. Ed. by Kang Zhang. – Boston; Dordrecht; London: Kluwer Academic Publ., 2003. – 442 p.
15. Diehl S. Software visualization: visualizing the structure, behaviour, and evolution of software. – Berlin: Springer, 2007. – 187 p.
16. Reiss S. The paradox of software visualization // *Proc. of 3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT)*. – Budapest, 2005. – P. 1–5.
17. Marcus A., Feng L., Maletic J.I. 3D representations for software visualization // *Proc. of the 2003 ACM symposium on Software visualization*. – San Diego, California, USA, 2003. – P. 27–36.
18. Teyseyre A., Campo M. An overview of 3D software visualization // *IEEE Transactions on Visualization and Computer Graphics*. – January–February 2009. – V. 15. – Iss. 1. – P. 87–105.
19. Panas T., Berrigan R., Grundy J. A 3D metaphor for software production visualization // *YLEM Journal: Artists Using Science and Technology*. – Spring 2005. – V. 25. – № 4. – P. 7–11.
20. Wetzel R., Lanza M. Visualizing software systems as Cities // *Proc. of Visualizing Software for Understanding and Analysis. VISSOFT 2007. 4th IEEE International Workshop*. – Banff, Canada, 2007. – P. 92–99.
21. Steinbrückner F., Lewerentz C. Representing development history in software Cities // *Proc. of the 5th international symposium on Software visualization (SOFTVIS '10)*. – Salt Lake City, Utah, USA, 2010. – P. 193–202.
22. Wetzel R., Lanza M., Robbes R. Software systems as cities: a controlled experiment // *Proc. of 33rd International Conference on Software Engineering (ICSE)*. – Honolulu HI USA, 2011. – P. 551–560.
23. Balogh G., Szabolics A., Beszédes A. CodeMetropolis: eclipse over the city of source code // *Proc. of IEEE 15th International Working Conference on Source Code Analysis and Manipulation, SCAM 2015*. – Bremen, Germany, 2015. – P. 271–276.
24. Авербух В.Л., Бакхтерев М.О., Манакон Д.В. Evaluations of visualization metaphors and views in the context of execution traces and call graphs // *Scientific Visualization*. – 2017, Quarter 4. – V. 9. – № 5. – P. 1–18.
25. Reed D., Scullin W., Tavera L., Shields K., Elford Ch. Virtual reality and parallel systems performance analysis. *IEEE Computer*, November 1995, vol. 28, no. 11. – P. 57–67.
26. Авербух Н.В. Психологические аспекты феномена присутствия в виртуальной среде // *Вопросы психологии*. – 2010. – № 5. – С. 105–113.
27. Knight C., Munro M. Virtual but Visible Software // *Proc. of IV IEEE International Conference Information Visualization*. – London, England, 2000. – P. 198–205.
28. Maletic J., Leigh J., Marcus A. Visualizing software in an immersive virtual reality environment // *Proc. of the ICSE'01 Workshop on Software Visualization*. – Toronto, Canada, 2001. – P. 49–54.
29. Glander T., Döllner J. Abstract representations for interactive visualization of virtual 3D city models // *Computers, Environment and Urban Systems*. – 2009. – V. 33. – Iss. 5. – P. 375–387.
30. Glander T., Döllner J. Automated cell-based generalization of virtual 3D city models with dynamic landmark highlighting // *Proc. of 11th ICA Workshop on Generalisation and Multiple Representations*. – Montpellier, France, 2008. URL: https://hpi.de/fileadmin/user_upload/fachgebiete/doellner/publications/2008/GD08/paper_ICA_workshop.pdf (дата обращения 15.11.2019).
31. Fittkau F., Krause A., Hasselbring W. Exploring software cities in virtual reality // *2015 IEEE 3rd Working Conference on Software Visualization (VISSOFT)*. – Bremen, Germany, 2015. – P. 130–134.
32. Fittkau F., Koppenhagen E., Hasselbring W. Research perspective on supporting software engineering via physical 3D models // *2015 IEEE 3rd Working Conference on Software Visualization (VISSOFT)*. – Bremen, Germany, 2015. – P. 125–129.
33. Vincur J., Navrat P., Polasek I. VR City: software analysis in virtual reality environment // *2017 IEEE International Conference on Software Quality, Reliability and Security Companion*. – Prague, Czech Republic, 2017. – P. 509–516.
34. Merino L., Ghafari M., Anslow C., Nierstrasz O. CityVR: Gameful Software Visualization // *IEEE International Conference on Software Maintenance and Evolution (ICSME TD Track)*. – Shanghai, China, 2017. – P. 633–637.
35. Oberhauser R., Carsten L. Gamified virtual reality for program code structure comprehension // *The International Journal of Virtual Reality*. – 2017. – V. 17 (02). – P. 79–88.
36. Oberhauser R., Lecon C. Virtual reality flythrough of program code structures // *Proc. of the 19th ACM Virtual Reality International Conference (VRIC 2017)*. – Laval, France, 2017. URL: https://www.hs-aalen.de/uploads/publication/file/9240/VRIC17_VR-FTC_CR.pdf (дата обращения 15.11.2019).
37. Oberhauser R. Immersive coding: a virtual and mixed reality environment for programmers // *Proc. of the Twelfth International Conference on Software Engineering Advances (ICSEA 2017)*. – Athens, Greece, 2017. – P. 250–255.
38. Oberhauser R., Lecon C. Immersed in software structures: a virtual reality approach // *ACHI 2017. The Tenth International Conference on Advances in Computer-Human Interactions*. – Nice, France, 2017. – P. 181–186.
39. Schreiber A., Bruggemann M. Interactive visualization of software components with virtual reality headsets // *2017 IEEE Working Conference on Software Visualization (VISSOFT)*. – Shanghai, China, 2017. – P. 119–123.
40. DebugAR: mixed dimensional displays for immersive debugging of distributed systems P. Reipschläger, S. Gumhold, B.K. Ozkan, R. Majumdar, A.Sh. Mathur, R. Dachselt // *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal, Canada, 2018. URL: https://www.researchgate.net/publication/324669075_DebugAR_Mixed_Dimensional_Displays_for_Immersive_Debugging_of_Distributed_Systems (дата обращения 15.11.2019).
41. Merino L., Bergel A., Nierstrasz O. Overcoming issues of 3D software visualization through immersive augmented reality // *2018 IEEE Working Conference on Software Visualization (VISSOFT)*. – Madrid, Spain, 2018. – P. 54–64.
42. Ens B., Anderson F., Grossman T., Annett M., Irani P., Fitzmaurice G. Ivy: exploring spatially situated visual programming for authoring and understanding intelligent environments. *Proc. of the 43rd Graphics Interface Conference GI '17*. – Edmonton, Alberta, 2017. – P. 156–162.
43. Merino L., Fuchs J., Blumenschein M., Anslow C., Ghafari M., Nierstrasz O., Behrisch M., Keim D.A. On the impact of the medium in the effectiveness of 3D software visualizations // *Proc. on 2017 IEEE Working Conference on Software Visualization (VISSOFT)*. – Shanghai, China, 2017. – P. 11–21.
44. Shneiderman B. The eyes have it: a task by data type taxonomy for information visualizations // *Proc. of the IEEE Conference on Visual Languages*. – Boulder, CO, USA, September 3–6, 1996. – P. 336–343.

45. Maletic J.I., Marcus A., Collard M.L. A task oriented view of software visualization // International Workshop on Visualizing Software for Understanding and Analysis. – Paris, France, 2002. – P. 32–40.
46. Авербух В.Л., Коновалов А.В., Балакин В.Н., Гомон М.Е., Слепцов П.В. Метод наглядного отображения транспьютерных сетей // Транспьютерные системы и их применение: Тез. конф. – Домодедово, 3–6 окт. 1994. – М.: Институт прикладной математики Российской академии наук, 1994.
47. Borkin M.A. Perception, cognition, and effectiveness of visualizations with applications in science and engineering: Doctoral dissertation. – Cambridge, Massachusetts, 2014. URL: <http://nrs.harvard.edu/urn-3:HUL.InstRepos:12274335> (дата обращения 15.11.2019).
48. Леонтьев А.Н. Деятельность. Сознание. Личность. – М.: Политиздат, 1975. – 304 с.
49. Рубинштейн С.Л. Основы общей психологии. – СПб.: Питер, 2005. – 713 с.
50. Barthelmess P., Anderson K. A view of software development environments based on activity theory // Computer Supported Cooperative Work (CSCW). – March 2002. – V. 11. – Iss. 1–2. – P. 13–37.
51. Georg G. Activity theory and its applications in software engineering and technology. Literature search results and observations // Technical Report CS-11-101. 2011. URL: <http://www.cs.colostate.edu/TechReports/Reports/2011/tr11-101.pdf> (дата обращения 15.11.2019).
52. Захарова А.А., Шкляр А.В. Метафоры визуализации // Научная визуализация. – 2013. – Кв. 2. – Т. 5. – № 2. – С. 16–24.
53. Захарова А.А., Шкляр А.В. Информационные признаки задач визуализации // Научная визуализация. – 2015. – Кв. 2. – Т. 7. – № 2. – С. 73–80.
54. Захарова А.А., Шкляр А.В., Ризен Ю.С. Измеряемые характеристики визуализации // Научная визуализация. – 2016. – Кв. 1. – Т. 8. – № 1. – С. 95–107.
55. Авербух В. Семiotический подход к формированию теории компьютерной визуализации // Научная визуализация. – 2013. – Кв. 1. – Т. 5. – № 1. – С. 1–25.
56. Averbukh V.L. Interdisciplinary Approaches to Semiotics // Semiotic analysis of computer visualization. – Croatia: InTech. Rijeka, 2017. – Ch. 6. – P. 97–133.

Поступила 02.12.2019 г.

Информация об авторах

Авербух В.Л., кандидат технических наук, заведующий сектором компьютерной визуализации Института математики и механики им. Н.Н. Красовского УрО РАН; доцент кафедры вычислительной математики и компьютерных наук Института естественных наук и математики Уральского Федерального Университета им. первого Президента России Б.Н. Ельцина.

Авербух Н.В., младший научный сотрудник лаборатории мозга и нейрокognитивного развития Уральского гуманитарного института, Уральского Федерального Университета им. первого Президента России Б.Н. Ельцина.

Васёв П.А., младший научный сотрудник сектора компьютерной визуализации Института математики и механики им. Н.Н. Красовского УрО РАН.

Гвоздарев И.Л., аспирант кафедры вычислительной математики и компьютерных наук Института естественных наук и математики Уральского Федерального Университета им. первого Президента России Б.Н. Ельцина.

Левчук Г.И., аспирант кафедры вычислительной математики и компьютерных наук Института естественных наук и математики Уральского Федерального Университета им. первого Президента России Б.Н. Ельцина.

Мелкозёров Л.О., студент департамента математики, механики и компьютерных наук Института естественных наук и математики Уральского Федерального Университета им. первого Президента России Б.Н. Ельцина.

UDC 004.923

SOFTWARE VISUALIZATION OF GEOSPATIAL DATA BASED ON VIRTUAL REALITY SYSTEMS. SURVEY AND FUTURE DEVELOPMENTS

Vladimir L. Averbukh^{1,2},
averbukh@imm.uran.ru

Natalya V. Averbukh²,
natalya_averbukh@mail.ru

Pavel A. Vasev¹,
vasev@imm.uran.ru

Ilya L. Gvozdarev²,
ilyagvz@gmail.com

Georgy I. Levchuk²,
levchukgeorgy1@gmail.com

Leonid O. Melkozerov²,
leoghost@mail.ru

¹ N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences, 16, Sofia Kovalevskaya street, Ekaterinburg, 620990, Russia.

² Ural Federal University named after the first President of Russia B.N. Yeltsin, 19, Mira street, Ekaterinburg, 620002, Russia.

The relevance of the research. In the time of global digitalization, when such projects as Digital Earth become relevant, computer visualization is a key element of computer modelling both in academic research and in technology, especially in the fields of natural and mineral resources management, and geo-resources exploration. Computer visualization is also necessary for presenting the «big data», for instance, data accumulated in geological research of large spaces. Computer visualization as an independent branch of science has been developing for over thirty years now. Now, Virtual Reality is actively used in computer visualization. Virtual Reality can be used in the presentation of Earth Resources Data. It is possible to create three-dimension displays of graphic objects representing the elements of the landscape. One of its subbranches is software visualization. Software visualization is essential in presenting complex software systems related to design and development in different fields of technology, including energy efficient technologies.

Objects. The paper is dedicated to research and development in the field of software visualization based on virtual reality. These researches can be applied in a number of areas related to Geoinformatics, as well as in the field of exploration of georesources.

Methods: methods of computer visualization theory, in particular, the elements of visualization metaphor theory.

The aim of the research and development is to find those approaches that will enable efficient use of virtual reality in solving complex problems that software specialists face. This software may be related to large volumes of geospatial data.

Results. The paper provides a survey of the development of software visualization and description of projects in this field, developed over the past decades. It describes the projects of visual programming and software visualization systems, developed by the authors on the basis of virtual reality with the use of cosmic space metaphors and an expanded city metaphor. It reviews the psychological aspects of the human factor in the development of software visualization systems with the use of virtual reality, including the use of activity theory. The paper arises a problem of describing the activity of a software developer acting as a user of software visualization systems. Software development for real-life technical problems (for example, for geological engineering or geo-resources forecasting) requires specialization and personalization of visualization systems. The results of the conducted analysis allow us to conclude that we are now at the early stages of crucial research.

Key words:

Software for geoinformatics, software visualization, visual programming, virtual reality, visualization metaphors, human factor, activity theory, geospatial data.

The work was supported by the Act 211 of the Government of the Russian Federation, contract no. 02.A03.21.0006.

REFERENCES

1. Working Conference on Software Visualization Charter. Available at: <http://vissoft-conferences.dcc.uchile.cl/> (accessed: 15 November 2019).
2. Shu N.C. Visual programming languages: a perspective and dimension analysis. *Visual Languages*. New York, Plenum Publ. Corporation, 1986. pp. 11–34.
3. Shu N.C. *Visual programming*. New York, Van Nostrand Reinhold Company, 1988. 314 p.
4. Shu N.C. Visual programming: perspectives and approaches. *IBM System Journal*, 1989, vol. 22, no. 4, pp. 525–547.
5. Brown M. *Algorithm animation*. Cambridge, Massachusetts, The MIT Press, 1988. 200 p.
6. Kovalev A.A., Lizenko S.L. Graficheskiy interfeys predstavleniya algoritmov i programm [Graphical interface for presentation of al-

- gorithms and programs]. *Upravlyayushchie sistemy i mashiny*, 1988, no. 4, pp. 41–47.
7. Parondzhanov V.D. *Uchis pisat, chitat i ponimat algoritmy. Algoritmy dlya pravilnogo myshleniya. Osnovy algoritimizatsii* [Learn to write, read and understand algorithms. Algorithms for correct thinking. The Basics of Algorithmization]. Moscow, DMK Press, 2012. 520 p.
 8. Price B.A., Small I.S., Baecker R.M. A taxonomy of software visualization. *Proc. of the 25th Hawaii International Conference on System Sciences (HICSS)*. Hawaii, USA, 1992. Vol. 2, pp. 597–606.
 9. Price B.A., Baecker R.M., Small I.S. A principled taxonomy of software visualization. *Journal of Visual Languages and Computing*, 1993, vol. 4, no. 3, pp. 211–266.
 10. Ball Th., Eick S.G. Software visualization in the Large. *IEEE Computer*, April 1996, vol. 29, no. 4, pp. 33–43.
 11. Averbukh V.L. Vizualnoe programmirovaniye i vizualizatsiya programmirovaniya. Opredeleniya i klassifikatsiya [Visual programming and program visualization. Definitions and classification]. *Vtoraya mezhdunarodnaya konferentsiya. Kompyuternaya grafika v nauke i iskusstve. Grafikon-92* [Second International Conference. Computer Graphics in Science and Art/ Graficon-92]. Moscow, 28 September – 2 October 1992. pp. 48–50.
 12. Averbukh V.L. *Vizualizatsiya programmogo obespecheniya* [Software visualization]. Yekaterinburg, Mathematical and mechanical faculty. Ural State University, 1995. 168 p.
 13. Socha D., Bailey M.L., Notkin D. Voyeur: graphical views of parallel programs. *Proc. of the ACM SIGPLAN and SIGOPS Workshop on Parallel and Distributed Debugging*. Madison, Wisconsin, May 5–6 1988. Vol. 24, no. 1, pp. 206–215.
 14. *Software visualization. From theory to practice*. Ed. by Kang Zhang. Boston, Dordrecht, London, Kluwer Academic Publ., 2003. 442 p.
 15. Diehl S. *Software visualization: visualizing the structure. Behaviour and evolution of software*. Berlin, Springer, 2007. 187 p.
 16. Reiss S. The paradox of software visualization. *Proc. of 3rd IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT)*. Budapest, 2005. pp. 1–5.
 17. Marcus A., Feng L., Maletic J.I. 3D representations for software visualization. *Proc. of the 2003 ACM symposium on Software visualization*. San Diego, California, USA, 2003. pp. 27–36.
 18. Teyseyre A., Campo M. An overview of 3D software visualization. *IEEE Transactions on Visualization and Computer Graphics*, January–February 2009, vol. 15, Iss. 1, pp. 87–105.
 19. Panas T., Berrigan R., Grundy J. A 3D metaphor for software production visualization. *YLEM Journal: Artists Using Science and Technology*, Spring 2005, vol. 25, no. 4, pp. 7–11.
 20. Wetzel R., Lanza M. Visualizing software systems as Cities. *Proc. of Visualizing Software for Understanding and Analysis. VISSOFT 2007. 4th IEEE International Workshop*. Banff, Canada, 2007. pp. 92–99.
 21. Steinbrückner F., Lewerentz C. Representing development history in software Cities. *Proc. of the 5th international symposium on Software visualization (SOFTVIS '10)*. Salt Lake City, Utah, USA, 2010. pp. 193–202.
 22. Wetzel R., Lanza M., Robbes R. Software systems as cities: a controlled experiment. *Proc. of 33rd International Conference on Software Engineering (ICSE)*. Honolulu HI USA, 2011. pp. 551–560.
 23. Balogh G., Szabolics A., Beszedes A. CodeMetropolis: Eclipse over the city of source code. *Proc. of IEEE 15th International Working Conference on Source Code Analysis and Manipulation. SCAM 2015*. Bremen, Germany, 2015. pp. 271–276.
 24. Averbukh V.L., Bakhterev M.O., Manakov D.V. Evaluations of visualization metaphors and views in the context of execution traces and call graphs. *Scientific Visualization*, 2017, Quarter 4, vol. 9, no. 5, pp. 1–18.
 25. Reed D., Scullin W., Tavera L., Shields K., Elford Ch. Virtual reality and parallel systems performance analysis. *IEEE Computer*, November 1995, vol. 28, no. 11, pp. 57–67.
 26. Averbukh N.V. Psychological aspects of the phenomenon of presence in the virtual environment. *Questions of psychology*, 2010, no. 5, pp. 105–113. In Rus.
 27. Knight C., Munro M. Virtual but visible software. *Proc. of IV IEEE International Conference Information Visualization*. London, England, 2000. pp. 198–205.
 28. Maletic J., Leigh J., Marcus A. Visualizing software in an immersive virtual reality environment. *Proc. of the ICSE'01 Workshop on Software Visualization*. Toronto, Canada, 2001. pp. 49–54.
 29. Glander T., Döllner J. Abstract representations for interactive visualization of virtual 3D city models. *Computers. Environment and Urban Systems*, 2009, vol. 33, Iss. 5, pp. 375–387.
 30. Glander T., Döllner J. Automated cell-based generalization of virtual 3D city models with dynamic landmark highlighting. *Proc. of 11th ICA Workshop on Generalisation and Multiple Representations*. Montpellier, France, 2008. Available at: https://hpi.de/fileadmin/user_upload/fachgebiete/doellner/publications/2008/GD08/paper_ICA_workshop.pdf (accessed 15 November 2019).
 31. Fittkau F., Krause A., Hasselbring W. Exploring software Cities in virtual reality. *2015 IEEE 3rd Working Conference on Software Visualization (VISSOFT)*. Bremen, Germany, 2015. pp. 130–134.
 32. Fittkau F., Koppenhagen E., Hasselbring W. Research perspective on supporting software engineering via physical 3D models. *2015 IEEE 3rd Working Conference on Software Visualization (VISSOFT)*. Bremen, Germany, 2015. pp. 125–129.
 33. Vincur J., Navrat P., Polasek I. VR City: software analysis in virtual reality environment. *2017 IEEE International Conference on Software Quality. Reliability and Security Companion*. Prague, Czech Republic, 2017. pp. 509–516.
 34. Merino L., Ghafari M., Anslow C., Nierstrasz O. CityVR: gameful software visualization. *IEEE International Conference on Software Maintenance and Evolution (ICSME TD Track)*. Shanghai, China, 2017. pp. 633–637.
 35. Oberhauser R., Carsten L. Gamified virtual reality for program code structure comprehension. *The International Journal of Virtual Reality*, 2017, vol. 17 (02), pp. 79–88.
 36. Oberhauser R., Lecon C. Virtual reality flythrough of program code structures. *Proc. of the 19th ACM Virtual Reality International Conference (VRIC 2017)*. Laval, France, 2017. Available at: https://www.hs-aalen.de/uploads/publication/file/9240/VRIC17_VR-FTC_CR.pdf (accessed 15 November 2019).
 37. Oberhauser R. Immersive coding: a virtual and mixed reality environment for programmers. *Proc. of the Twelfth International Conference on Software Engineering Advances (ICSEA 2017)*. Athens, Greece, 2017. pp. 250–255.
 38. Oberhauser R., Lecon C. Immersed in software structures: a virtual reality approach. *ACHI 2017. The Tenth International Conference on Advances in Computer-Human Interactions*. Nice, France, 2017. pp. 181–186.
 39. Schreiber A., Bruggemann M. Interactive visualization of software components with virtual reality headsets. *2017 IEEE Working Conference on Software Visualization (VISSOFT)*. Shanghai, China, 2017. pp. 119–123.
 40. Reipschläger P., Gumhold S., Ozkan B.K., Majumdar R., Mathur A.Sh., Dachselt R. DebugAR: mixed dimensional displays for immersive debugging of distributed systems. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal, Canada, 2018. Available at: https://www.researchgate.net/publication/324669075_DebugAR_Mixed_Dimensional_Displays_for_Immersive_Debugging_of_Distributed_Systems (accessed 15 November 2019).
 41. Merino L., Bergel A., Nierstrasz O. Overcoming issues of 3D software visualization through immersive augmented reality. *2018 IEEE Working Conference on Software Visualization (VISSOFT)*. Madrid, Spain, 2018. pp. 54–64.
 42. Ens B., Anderson F., Grossman T., Annett M., Irani P., Fitzmaurice G. Ivy: exploring spatially situated visual programming for authoring and understanding intelligent environments. *Proc. of the 43rd Graphics Interface Conference GI '17*. Edmonton, Alberta, 2017. pp. 156–162.
 43. Merino L., Fuchs J., Blumenschein M., Anslow C., Ghafari M., Nierstrasz O., Behrisch M., Keim D.A. On the impact of the medium in the effectiveness of 3D software visualizations. *Proc. on 2017 IEEE Working Conference on Software Visualization (VISSOFT)*. Shanghai, China, 2017. pp. 11–21.
 44. Shneiderman B. The eyes have it: a task by data type taxonomy for information visualizations. *Proc. of the IEEE Conference on Visu-*

- al Languages*. Boulder, CO, USA, September 3–6 1996. pp. 336–343.
45. Maletic J.I., Marcus A., Collard M.L. A task oriented view of software visualization. *International Workshop on Visualizing Software for Understanding and Analysis*. Paris, France, 2002. pp. 32–40.
 46. Averbukh V.L., Konovalov A.V., Balakin V.N., Gomon M.E., Sleptsov P.V. Metod naglyadnogo otobrazheniya transpyuternykh setey [The method of visual display of transputer networks]. *Transpyuternye sistemy i ikh primeneniye* [Transputer systems and their application]. Domodedovo, 3–6 October 1994. Moscow, Institute of Applied Mathematics of the Russian Academy of Sciences, 1994.
 47. Borkin M.A. *Perception, cognition and effectiveness of visualizations with applications in science and engineering*. Doctoral dissertation. Cambridge, Massachusetts, 2014. Available at: <http://nrs.harvard.edu/urn-3:HUL.InstRepos:12274335> (accessed 15 November 2019).
 48. Leontyev A.N. *Deyatelnost. Soznaniye. Lichnost* [Activity. Consciousness. Personality]. Moscow, Politizdat Publ., 1975. 304 p.
 49. Rubinshteyn S.L. *Osnovy obshchey psikhologii* [Basics of general psychology]. St-Petersburg, Piter Publ., 2005. 713 p.
 50. Barthelme P., Anderson K. A view of software development environments based on activity theory. *Computer Supported Cooperative Work (CSCW)*, March 2002, vol. 11, Iss. 1–2, pp. 13–37.
 51. Georg G. Activity theory and its applications in software engineering and technology. Literature search results and observations. *Technical Report CS-11-101*. 2011. Available at: <http://www.cs.colostate.edu/TechReports/Reports/2011/tr11-101.pdf> (accessed 15 November 2019).
 52. Zakharova A.A., Shklyar A.V. Metafory vizualizatsii [Visualization metaphors]. *Nauchnaya vizualizatsiya*, 2013, Q. 2, vol. 5, no. 2, pp. 16–24.
 53. Zakharova A.A., Shklyar A.V. Informatsionnye priznaki zadach vizualizatsii [Informative features of data visualization tasks]. *Nauchnaya vizualizatsiya*, 2015, Q. 2, vol. 7, no. 2, pp. 73–80.
 54. Zakharova A.A., Shklyar A.V., Rizen Yu.S. Izmeryaemye kharakteristiki vizualizatsii [Measurable features of visualization tasks]. *Nauchnaya vizualizatsiya*, 2016, Q. 1, vol. 8, no. 1, pp. 95–107.
 55. Averbukh V., Semioticheskiy podkhod k formirovaniyu teorii kompyuternoy vizualizatsii [Semiotic approach to forming the theory of computer visualization]. *Nauchnaya vizualizatsiya*, 2013, Q. 1, vol. 5, no. 1, pp. 1–25.
 56. Averbukh V.L. *Semiotic analysis of computer visualization*. Ch. 6. Interdisciplinary approaches to semiotics. Croatia, InTech. Rijeka, 2017. pp. 97–133.

Received: 2 December 2019.

Information about the authors

Vladimir L. Averbukh, Cand. Sc., head of computer visualization sector, N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences; associate professor, Ural Federal University named after the first President of Russia B.N. Yeltsin.

Natalya V. Averbukh, junior researcher, Ural Federal University named after the first President of Russia B.N. Yeltsin.

Pavel A. Vasev, junior researcher, N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences.

Ilya L. Gvozdev, postgraduate student, Ural Federal University named after the first President of Russia B.N. Yeltsin.

Georgy I. Levchuk, postgraduate student, Ural Federal University named after the first President of Russia B.N. Yeltsin.

Leonid O. Melkozerov, student, Ural Federal University named after the first President of Russia B.N. Yeltsin.