

Визуализация программного обеспечения с использованием виртуальной реальности

В.Л. Авербух^{1,2}, И.Л. Гвоздарев², Г.И. Левчук²
averbukh@imm.uran.ru, ilyagvz@gmail.com, levchukgeorgiy1@gmail.com

¹ ИММ УрО РАН, ² УрФУ, г. Екатеринбург, Российская Федерация

Работа посвящена системам визуализации программного обеспечения на базе виртуальной реальности. Делается обзор разработки таких систем. Описываются проекты систем визуального программирования и визуализации программного обеспечения на базе виртуальной реальности с использованием метафор солнечной системы и расширенной метафоры города.

Ключевые слова: визуализация программного обеспечения, виртуальная реальность, метафоры визуализации.

Software Visualization based on Virtual Reality

V. L. Averbukh, I. L. Gvozdarov, G. I. Levchuk

The paper is devoted to Software Visualization systems based on Virtual Reality. The overview of design and development of such systems is presented. The projects of visual programming and software visualization systems based on virtual reality using Solar System metaphor and the extended City metaphor are described.

Keywords: software visualization, virtual reality, visualization metaphors.

1. Введение

Работа посвящена проблематике визуализации программного обеспечения и современному состоянию этой дисциплины в связи с использованием сред виртуальной реальности. Визуализация программного обеспечения (Software Visualization), которая активно развивается уже более тридцати лет, является подобластью дисциплины Компьютерная визуализация. Этой теме посвящено большое количество публикаций и выступлений на различных конференциях. Ежегодно проводятся конференции по визуализации программного обеспечения *VISSOFT*. В последнее время вновь проявился интерес к использованию сред виртуальной реальности в системах визуализации программного обеспечения. Ниже будут описаны некоторые положения этой дисциплины. На примере метафоры города рассматривается понятие *свойство метафор*. Затем приводится обзор систем визуализации программного обеспечения на базе сред виртуальной реальности. Анализ этих систем показывает необходимость проектирования и разработки собственных прототипов и проведения исследований возникающих возможностей и проблем.

2. Визуализация программного обеспечения

Под визуализацией программного обеспечения понимается совокупность методик использования графики и средств человеко-машинного взаимодействия, применяемых для лучшего уяснения понятий и эффективной эксплуатации программного обеспечения, а также для спецификации и представления программных объектов в процессе создания программ. Первые системы визуального программирования и визуализации программ, которые можно отнести к визуализации программного обеспечения, появились в конце 70-ых — начале 80-ых годов. Появление этой дисциплины связано с практикой и идеями графического представления программ, основанными на активном использовании программистами блок-схем (графов потоков управления) и графов потоков данных. В число систем визуализации программного обеспечения обычно включаются системы визуального программирования, системы визуализации программирования. Выдвигалась идея, что и разработка программ, и их отладка, и их сопровождение должны

осуществляться в рамках единой системы с одинаковым графическим представлением программных сущностей. Описание новой дисциплины было осуществлено в начале 90-ых. Несмотря на использование примитивной (по нынешним временам) графики в 80-ых и 90-ых годах были разработаны интересные системы визуального программирования и полноценные визуальные отладчики. Однако использование традиционных методов двумерной графики очень быстро стало ограничением для систем визуализации программного обеспечения. Даже сравнительно небольшие отображения, например, сотни связанных между собой объектов были сложны как для восприятия, так и для интерпретации. Появились публикации, в которых указывалось на кризис в области визуализации программного обеспечения. Выходом в этом случае стало использование метафор визуализации, трехмерной графики и средств виртуальной реальности.

3. Метафоры визуализации

Неформально под метафорой визуализации понимается главная идея при отображении прикладной области на визуальный мир. Метафоры используются для определения 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. На этом примере мы рассмотрим такое понятие как «свойство метафоры». Метафора города дает возможность получить представление структуры программного проекта в виде карты некоторой местности, реалистичного трёхмерного представления городских кварталов, завода или обобщенных образов, напоминающих фантастический пейзаж. Возможности этой метафоры позволяют отображать *иерархию элементов системы* (отдельные строения в рамках городского квартала), *потоки управления, потоки данных, иные связи между элементами программ, программными конструкциями или частями программного комплекса* (улицы и дороги, связывающие цеха фабрики). Эта метафора может служить основой для построения видов отображения в системах отладки, тестирования и мониторинга работы параллельных и распределенных программ, также как и программ, обрабатывающих события и обеспечивающих реакции на

них. Среди свойств метафоры города можно выделить **неограниченный контекст, естественность, организация внутренней структуры, устойчивость к масштабированию**. Метафора города предполагает наличие внутренней структуры и допускает представление достаточно большого объема однородной в визуальном смысле информации. При этом интерпретация полученных графических выводов представляется простой. Возможность пролета над городом позволяет осуществление удобной и легкой навигации. Полет с изменением высоты дает возможность реализовать масштабирование и зуминг. Это позволяет использовать метафору города при разработке систем визуализации программного обеспечения на базе *сред виртуальной реальности*.

4. Визуализация программного обеспечения на базе сред виртуальной реальности

В первой половине 90-ых годов появилась первая система визуализации программного обеспечения, активно использующая средства виртуальной реальности и метафору комнаты-здания в трехмерном варианте и функционировала на базе среды виртуальной реальности CAVE. Визуальная среда Avatar [12] была предназначена для представления больших объемов данных о производительности параллельных систем, получаемых непосредственно в ходе их работы, что использовалось для отладки производительности параллельных программ. По ходу работы пользователь как бы оказывается внутри трехмерного помещения, где на стены проецируется видеоизображение. На внутренних гранях куба - полу и стенах помещения выводятся оси и кривые, описывающие метрики производительности параллельной программы. Данные о производительности параллельных программ отображались в виде традиционных двумерных графиков. Реализовано объединение отдельных элементов, которое напоминает стеклянный небоскреб. Вводится визуальное отображение - аналог фазового портрета - "лента истории". Путешествие ("виртуальный полет") по небоскребу на основе этой ленты давал возможность исследования данных о производительности параллельной программы. Обратим внимание на то, что несмотря на очень интересную реализацию системы Avatar сведений о ее использовании или продолжении разработок найти не удалось, что возможно связано как с недостаточной информативностью двумерных графиков, использованных при визуализации данных о производительности процессоров, так и с возможными неприятными и даже болезненными ощущениями проявлениями, вызванными у пользователя при "виртуальном полете", который проходит согласно ленте истории работы многопроцессорной системы. Проявления *киберболезни (cyberthickness)* часто связываются с отсутствием возможности активного управления событиями виртуальной реальности.

Новый интерес к использованию виртуальной реальности в системах визуализации программного обеспечения проявился в самом начале 2000-ых годов [4], [5], хотя дальше интересных идей первые публикации нового тысячелетия не пошли. Вновь работы в этом направлении начали появляться несколько лет назад.

В работе [3] было описан прототип системы визуализации программного обеспечения на основе метафоры города, в которой использовался шлем виртуальной реальности и интерфейс 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. Также эта исследовательская команда опубликовала работу по использованию физического моделирования на базе 3D-принтинга для задач визуализации программного обеспечения. [2] 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 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. Две системы, известные из публикаций 2017 года, носят весьма схожие имена - VR City [15] и CityVR [7]. 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 [7] рассматривается не только использование метафоры города для визуализации программного обеспечения на базе виртуальной реальности, но и подход, который называется *геймификация* задач разработки программного обеспечения. Этот подход подразумевает создание инструментов, которые предоставляют разработчикам программного обеспечения интерфейс, аналогичный компьютерным играм. Проводился анализ того, как разработчики взаимодействуют с системой визуализации CityVR. Разработчики были взволнованы, они почувствовали не только интерес к работе, но и определенный вызов. Они испытывали погружение в виртуальный мир при сохранении управляемости системы. При взаимодействии для навигации в виртуальном мире и при выборе нужных элементов программы было затрачено значительное время, при этом пользователи считали, что время прошло быстрее, чем в реальности. Также на *геймификацию* при разработке систем визуализации программного обеспечения на базе виртуальной реальности указывается в [11]. Разрабатываемая среда на базе виртуальной реальности 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. Эта команда исследователей опубликовала несколько работ, посвященных разработке систем визуализации программного обеспечения на базе виртуальной реальности [9],[10] (включая работу по программированию с использованием виртуальной реальности [8]). В работе [9] рассматривается перспективы использования погружения (immersion) разработчиков в структуры программ, а в [10] виртуальный полет над программными объектами в рамках различных метафор визуализации. Работа [14] 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 [13] 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. В [8] представлена система на базе виртуальной реальности для программистов. Эта система должна обеспечить визуализацию структуры программного кода в нескольких метафорах. Реализован встроенный просмотр работы с реальными клавиатурой и мышью и взаимодействие в смешанной реальности. Это обеспечивает поддержку основных задач программирования, не выходя из среды виртуальной реальности. Реализован прототип системы. Проведено исследование, демонстрирующее осуществимость идей. Получены результаты эмпирической оценки, демонстрирующие потенциал системы.

В работе [1] 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.

Наконец отметим работу, посвященную оценке эффективности систем трехмерной визуализации программного обеспечения [6]. Для эксперимента была реализована трехмерная визуализация на базе метафоры города на стандартном экране компьютера, в среде виртуальной реальности и на 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 наименьшее 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.

5. Разработка прототипов систем

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. Ниже описываются проекты и реализации прототипов систем визуализации программного обеспечения на базе сред виртуальной реальности, которые послужат для этих нужд.

Разрабатывается визуальный язык на основе объектов, когда визуальная среда программирования представляет графические или символичные элементы, которыми можно манипулировать в соответствии с некоторыми правилами. Целью проекта является создание среды визуального

программирования с возможностью абстрагироваться от правил и спецификаций конкретного текстового языка, чтобы 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 (несоответствие типов, нарушение синтаксиса и т.п.).

В качестве идеи среды визуального программирования выбрана метафора солнечной системы, среди свойств которой можно указать на *естественность, организация внутренней структуры, устойчивость к масштабированию*. Данную метафору можно использовать при разработке систем визуализации программного обеспечения на базе *сред виртуальной реальности*. Все пользовательские классы представляются планетами. (Рис. 1). 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. Использование виртуальной реальности дает преимущества в навигации. Появляется возможность путешествия по структуре классов, что поможет устранить неоднозначности сложных абстрактных представлений, позволит пользователю оценить относительный размер объектов и дистанции между ними. Предполагается осуществлять взаимодействие с системой с помощью используемых в средах виртуальной реальности устройств, а не традиционной клавиатуры. Реализована возможность создания примитивных классов, внутри которых нет зависимости от других классов (поля-примитивы), с помощью манипулирования графическими объектами. Также реализована возможность создания циклов и условных операторов с помощью манипулирования графическими объектами. Увеличение сложности ПО не должно сильно увеличивать сложность графического представления. В дальнейшем возможно расширение среды визуальной разработки ПО.

Кроме того, разрабатывается проект визуализации программного обеспечения для нужд разработчиков и тестировщиков на базе виртуальной реальности. Предполагается использование двух форм представления — графовое и на основе метафоры города. В результате обеспечивается взаимодействие с пользователем при формировании визуализации выбранного проекта с выведением информации о всех составляющих единицах кода. Представляется, что для тестировщиков удобнее визуализация на базе метафоры города. Разработчик, уяснив проблему, открывают визуализацию кода в метафоре графа. Графовое представление используется для исследования кода и получения информации о его структуре на данный момент, а метафора города позволяет получить информацию для коррекции кода. (Рис. 2). Разрабатывается функционал для добавления класса в виде трехмерной модели здания, что позволит использовать внутренние элементы здания для визуализации класса. 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 в каких

местах они используются, где изменяются и как проходит процесс работы алгоритмов. 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 внутри объектов, размещать в трехмерной модели различные датчики и увидеть, как функционирует устройство под управлением программы.

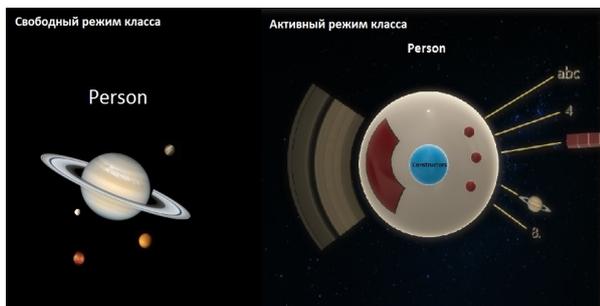


Рис. 1. Общий вид среды визуального программирования на основе метафоры солнечной системы.

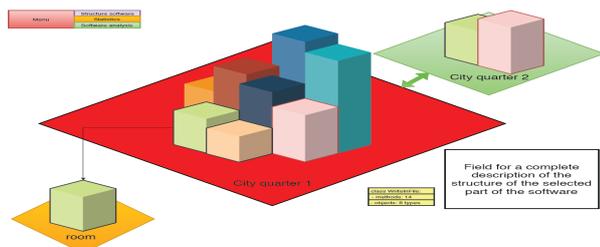


Рис. 2. Общий вид интерфейса системы на базе метафоры города

6. Заключение

Разработка прототипов систем на базе виртуальной реальности — лишь первый этап исследований. На базе этих систем можно прояснить те вопросы, на которые нельзя найти ответы в имеющихся публикациях. Необходим поиск критериев выбора метафор визуализации и подходов к проектированию видов отображения. Не ясно насколько эффективны среды виртуальной реальности для использования в 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 весьма интересна, но не всегда может быть использована в серьезных разработках. Отсюда вытекает проблема описания деятельности программиста в качестве пользователя систем визуализации программного обеспечения. Мы занимаемся изучением деятельности пользователей систем научной визуализации. Причем эта задача требует большого числа экспериментов и соответствующей программной поддержки. Отметим, что упомянутые в приведенном выше обзоре физическая визуализация и жестовые интерфейсы использовались нами в случае специализированных систем научной визуализации. Причем физическая визуализация моделей оказалась весьма интересной и полезной для конечных пользователей, а простой язык жестов оказался достаточно эффективным. Для систем визуализации

программного обеспечения необходимы новые исследования и экспериментальные разработки. Наконец, при проектировании систем на базе виртуальной реальности следует учитывать такой фактор как феномен присутствия (*presence*), а также проявления киберболезни, тем более, что до 30% участников исследований могут почувствовать ее проявление [9].

Литература

- [1] Ens B., Anderson F., Grossman T., Annett M., Irani P., Fitzmaurice G. Ivy: Exploring Spatially Situated Visual Programming for Authoring and Understanding Intelligent Environments // Proceedings of the 43rd Graphics Interface Conference GI '17. 2017. Pp. 156-162.
- [2] Fittkau F., Koppenhagen E., Hasselbring W. Research Perspective on Supporting Software Engineering via Physical 3D Models // 2015 IEEE 3rd Working Conference on Software Visualization (VISOFT). Pp. 125-129.
- [3] Fittkau F., Krause A., Hasselbring W. Exploring Software Cities in Virtual Reality // 2015 IEEE 3rd Working Conference on Software Visualization. Pp. 130-134.
- [4] Knight C., Munro M. Virtual but Visible Software // IV 2000 Information Visualization. IEEE International Conference on Information Visualization, 2000. Proceedings. Pp. 198-205.
- [5] Maletic J., Leigh J., Marcus A. Visualizing Software in an Immersive Virtual Reality Environment // Proceedings of the Workshop on Software Visualization. 2001. Pp. 49-54.
- [6] 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 // Proceedings on 2017 IEEE Working Conference on Software Visualization, 2017. Pp. 11-21.
- [7] Merino L., Ghafari M., Anslow C., Nierstrasz O. CityVR: Gameful Software Visualization // IEEE International Conference on Software Maintenance and Evolution (ICSME TD Track). 2017. Pp. 633-637.
- [8] Oberhauser R. Immersive Coding: A Virtual and Mixed Reality Environment for Programmers // Proceedings of The Twelfth International Conference on Software Engineering Advances (ICSEA 2017). Pp. 250-255.
- [9] Oberhauser R. Carsten Lecon Immersed in Software Structures: A Virtual Reality Approach // ACHI 2017. The Tenth International Conference on Advances in Computer-Human Interactions. Pp. 181-186.
- [10] Oberhauser R. Carsten L. Virtual Reality Flythrough of Program Code Structures // Proc. of the 19th ACM Virtual Reality International Conference. ACM, 2017. (4 pp.)
- [11] Oberhauser R. Carsten L. Gamified Virtual Reality for Program Code Structure Comprehension // The International Journal of Virtual Reality, 2017, 17 (02). Pp. 79-88.
- [12] Reed D., Scullin W., Tavera L., Shields K., Elford Ch. Virtual Reality and Parallel Systems Performance Analysis // IEEE Computer, V.28, N 11, (1995). Pp. 57-67.
- [13] 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. (6 pp.)
- [14] Schreiber A., Bruggemann M. Interactive Visualization of Software Components with Virtual Reality Headsets // 2017 IEEE Working Conference on Software Visualization (VISOFT), 2017. Pp. 119-123.
- [15] 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. Pp. 509 – 516.

Об авторах

Авербух Владимир Лазаревич, заведующий сектором компьютерной визуализации ИММ УрО РАН; доцент кафедры вычислительной математики и компьютерных наук УрФУ. E-mail – averbukh@imm.uran.ru.

Гвоздарев Илья Леонидович, аспирант кафедры вычислительной математики и компьютерных наук УрФУ E-mail – ilyagvz@gmail.com.

Левчук Георгий Иванович, аспирант кафедры вычислительной математики и компьютерных наук УрФУ E-mail – levchukgeorgy1@gmail.com.