

## Новые возможности адаптации систем научной визуализации к сторонним решателям\*

К.В. Рябинин, С.И. Чуприна, А.Ю. Бортников

kostya.ryabinin@gmail.com | chuprinas@inbox.ru | a.yu.bortnikov@yandex.ru

Россия, Пермский государственный национальный исследовательский университет

*В данной статье рассматриваются новые средства, разработанные в рамках ранее предложенной авторами концепции адаптации систем научной визуализации к сторонним решателям: аппарат семантических фильтров и тонкий браузерный клиент. Эти средства внедрены в мультиплатформенную клиент-серверную систему научной визуализации SciVi, имеющую высокоуровневые средства настройки на специфику предметной области решателей и поддерживающую автоматическую конвертацию данных, полученных от решателей, в пригодный для рендеринга формат.*

*Сервер системы SciVi предназначен для работы на настольных компьютерах; клиенты реализованы в виде нативных приложений на основе OpenGL / OpenGLES и в виде Web-приложений на основе WebGL. Поддержка мультиплатформенности с сохранением высокой эффективности визуализации осуществляется путём автоматической адаптации к конкретным особенностям программно-аппаратной платформы по заданным эвристическим правилам.*

*Расширяемость и гибкость достигается за счёт управления работой системы через онтологическую базу знаний, включающую онтологию синтаксических конструкций ввода-вывода языков программирования, онтологию графических объектов и онтологию семантических фильтров. Для осуществления настройки системы визуализации на специфику конкретной задачи пользователю достаточно провести связи между узлами, описывающими входные и выходные данные решателя, и выбранными для визуализации объектами. В качестве промежуточных узлов полученной в результате этого диаграммы потока данных выступают семантические фильтры, предлагаемые пользователю на выбор из репозитория системы.*

**Ключевые слова:** научная визуализация, мультиплатформенность, онтологический инжиниринг, мобильные устройства, поток данных, тонкий клиент, WebGL

### Введение

Большинство современных систем научной визуализации универсального назначения (таких, например, как TecPlot, Avizo, VizIt, ParaView, KiwiViewer и др.) имеют достаточно жёстко регламентированные форматы входных данных, не предоставляя при этом никаких средств для высокоуровневой адаптации к специфике сторонних источников данных. Для пользователей это зачастую означает необходимость ручной настройки источников данных на требуемый визуализаторами формат. Обычно такими источниками служат программные или программно-аппаратные системы (решатели, англ. Solvers), а также хранилища данных. В этом случае от пользователя требуется либо изменять программный код решателя так, чтобы генерируемые им данные удовлетворяли требованиям визуализатора, либо создавать промежуточное программное обеспечение для конвертации. Это, в свою очередь, требует дополнительных трудозатрат и наличия определённой программистской квалификации, затрудняя процесс автоматизированного создания высококачественных изображений, удовлетворяющих индивидуальным потребностям исследователей, и снижая в конечном итоге эффективность их научной работы.

Кроме того, среди популярных систем научной визуализации отсутствуют мультиплатформенные решения, позволяющие осуществлять высококачественный рендеринг интересующих пользователя данных как на настольных компьютерах, так и на мобильных устройствах. В контексте же работы в полевых условиях (например, в экспедициях) наличие мобильных версий систем визуального анализа научных данных является весьма актуальным.

Для комплексного решения указанных проблем авторами был предложен модельно-ориентированный подход к разработке систем научной визуализации, основанный на методах онтологического инжиниринга [6]. Суть этого подхода заключается в том, что работа системы научной визуализации полностью управляется знаниями об отображаемых графических объектах, о специфике источников данных и о программно-аппаратных особенностях вычислительной среды. Знания хранятся в виде онтологий – формальных моделей предметных областей, включающих в себя множество понятий этих предметных областей с их определениями, множество связей между понятиями и множество аксиом, описывающих семантические ограничения и другие правила, вводимые для соответствующих понятий и связей.

Использование принципов управляемости системы визуализации знаниями о её визуальных средствах, особенностях источников данных и окружающей

Работа опубликована при финансовой поддержке РФФИ, грант 16-07-20482.

вычислительной среды позволяет во многом упростить и автоматизировать следующие ключевые процессы разработки и использования системы:

1. Расширение новыми графическими возможностями путём добавления поддержки отображения новых видов объектов и новых типов графических сцен.
2. Настройку на специфику решаемой задачи и конкретные источники данных без модификации способа представления данных на стороне источников.
3. Организацию обратной связи с источниками данных (например, автоматизированную генерацию графического интерфейса на стороне клиента для управления вычислительным процессом решателя напрямую из системы визуализации, без необходимости модификации исходного кода решателя).
4. Поддержку мультиплатформенности с сохранением высокой эффективности рендеринга путём адаптации к особенностям конкретной программно-аппаратной платформы.

На базе предложенного подхода авторами была разработана универсальная мультиплатформенная клиент-серверная система научной визуализации SciVi [2], которая успешно использовалась для решения целого ряда задач рендеринга научных данных из различных предметных областей [3].

Архитектура системы SciVi нацелена на высокую эффективность визуализации и расширяемость функциональности. Целью данной работы является дальнейшее развитие системы SciVi в рамках предложенной концепции управляемости знаниями. Основные пути развития:

1. Разработка удобной и интуитивно-понятной пользователям подсистемы фильтрации данных.
2. Реализация тонкого Web-клиента для организации работы с системой без необходимости установки на компьютер пользователя дополнительного программного обеспечения.

### Общая характеристика системы научной визуализации SCIVI

Сервер системы SciVi предназначен для работы на настольном компьютере под управлением ОС Windows, GNU/Linux или OS X. Он отвечает за взаимодействие с источниками данных (чаще всего – решателями, которые являются сторонними по отношению к системе визуализации) и предобработку данных, получаемых из этих источников, с целью конвертации их к пригодному для визуализации в конкретных условиях виду.

Нативные клиенты системы SciVi реализованы как для настольных компьютеров под управлением Windows, GNU/Linux и OS X, так и для мобильных устройств под управлением iOS и Android. Запущенный на конкретном оборудовании клиент производит измерение своей производительности и сообщает результат измерения серверу. На основании

этих данных, руководствуясь набором эвристических правил, сервер автоматически переключает работу системы визуализации в один из трёх режимов: визуализация полного объёма данных на стороне клиента, частичное упрощение данных сервером и их визуализация на стороне клиента, визуализация полного объёма данных на стороне сервера и передача клиенту готового изображения.

В первой версии системы SciVi её база знаний включала в себя две онтологии: онтологию  $L$  синтаксических конструкций ввода-вывода языков программирования и онтологию  $U$  графических объектов и сцен, используемых для визуализации. Онтология  $L$  используется для автоматической генерации синтаксических анализаторов программного кода, предназначенных для извлечения структур входных и выходных данных поданных на анализ программ. Если у пользователя есть доступ (на чтение) к исходному коду решателя, процесс настройки системы визуализации на формат входных и выходных данных этого решателя полностью автоматизируется. В случае отсутствия исходного кода, пользователь должен описать структуру входных и выходных данных вручную при помощи высокоуровневого графического интерфейса.

Онтология  $U$  используется для описания поддерживаемых системой графических объектов и типов сцен. Пользователь при помощи высокоуровневого графического интерфейса выбирает соответствующие специфике решаемой задачи тип сцены и объекты, а также назначает соответствия элементов структуры входных и выходных данных решателя свойствам этих объектов.

При наличии исполняемого файла решателя возникает возможность настройки обратной связи системы SciVi с этим решателем так, чтобы в процессе визуализации пользователь мог при помощи автоматически сгенерированного графического интерфейса изменить входные данные и запустить процесс повторной генерации подлежащего визуализации результата (т. н. «режим on-line»). При отсутствии исполняемого файла решателя возможна визуализация заранее подготовленных данных, сохранённых в некотором хранилище, например, в файле (т. н. «режим off-line»).

### Подсистема семантических фильтров

Онтологии  $L$  и  $U$  используются для настройки системы SciVi на специфику предметной области и особенности решателя.

Однако часто возникает необходимость дополнительной настройки на специфику решаемой задачи, когда пользователю требуется осуществить некоторую особую обработку (фильтрацию) данных, выдаваемых решателем. Чтобы освободить пользователя от необходимости вносить изменения в решатель или создавать промежуточное программное

обеспечение для такого рода фильтрации, было решено реализовать механизм настраиваемой предобработки данных на стороне SciVi, добавив в её состав подсистему т. н. семантических фильтров.

Семантический фильтр – это отображение  $\varphi$  вида

$$\varphi : \langle I, S \rangle \rightarrow O$$

где  $I$  – множество типизированных входов,  $S$  – множество настроечных параметров,  $O$  – множество типизированных выходов.

Для представления семантических фильтров в системе SciVi введена дополнительная онтология  $F$ , описывающая допустимые типы данных и доступные фильтры, а также содержащая ссылки на библиотечные модули, реализующие эти фильтры.

Предобработка данных посредством семантических фильтров осуществляется в два единообразно представляемых этапа: первичная обработка на стороне сервера и окончательная – на стороне клиента.

На стороне сервера выполняется наиболее общая фильтрация, которая настраивается вместе с регистрацией в системе очередного решателя и, как правило, изменяется достаточно редко. Цель этой фильтрации – удовлетворить базовые потребности пользователя в изменении данных, порождаемых решателем. На этот этап предлагается выносить наиболее вычислительно сложные операции, так как чаще всего сервер превосходит клиентов по вычислительной мощности.

В процессе работы системы клиенту отправляется онтологический профиль данных, получаемых в результате первичной фильтрации. Этот профиль представляет собой онтологическое описание структуры и типов кортежей множества результирующих данных.

На стороне клиента выполняется фильтрация, адаптирующая получаемые им данные к решению частных задач. Настройка этой фильтрации может изменяться независимо от сервера.

Фактические данные, необходимые для визуализации, кешируются на стороне клиента, что даёт возможность решать различные частные задачи визуализации и анализа без участия сервера (в «режиме off-line»), только лишь путём перенастройки фильтрации. Например, пользователь может изменять линейный масштаб значений данных на логарифмический, или отсекал часть данных некоторой оконной функцией.

Настройка семантических фильтров для обоих этапов фильтрации осуществляется единообразно и представляет собой построение диаграммы потока данных (англ. *Data Flow*) [1] при помощи встроенного в систему SciVi визуального редактора графов. Вершины в этой диаграмме соответствуют

фильтрам, а дуги – связи по данным. Каждая вершина, в зависимости от представляемого ей фильтра, может иметь несколько входов и несколько выходов, а также несколько управляющих элементов (полей ввода значений, слайдеров, радиокнопок и т. п.), соответствующих параметрам этого фильтра. Пример диаграммы потока данных, формируемой на стороне клиента, представлен на рис. 1.

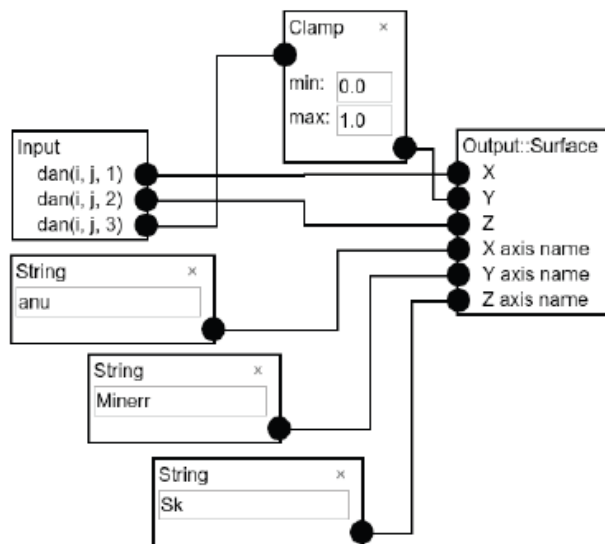


Рис. 1: Пример диаграммы потока данных, соответствующей настройке семантических фильтров на уровне графического интерфейса клиента системы научной визуализации SciVi

Согласно онтологическому профилю данных автоматически формируется вершина «Input». Далее пользователь выбирает из имеющейся палитры инструментов, автоматически сгенерированной на основе онтологии  $U$ , интересующий его тип графической сцены и визуальный объект. В результате этого создаётся вершина (или группа вершин) «Output», отражающая все свойства выбранного объекта и типа сцены. После этого пользователь может выбрать нужные фильтры из палитры, автоматически сгенерированной по онтологии  $F$ . В результате создаются соответствующие вершины. Особой категорией фильтров являются строковые, числовые и цветные константы, полезные, например, для настройки подписей осей на графиках, цветных шкал и т. п. Заключительным шагом настройки является соединение соответствующих входов и выходов имеющихся на графе вершин так, чтобы в итоге получилась интересующая пользователя схема алгоритма, который преобразует входные данные и определяет соответствующие параметры визуальных объектов и графической сцены. В приведённом на рис. 1 примере элемент подлежащих визуализации данных  $dan(i, j, 3)$  подвергается воздействию фильтра *Clamp*, который име-

ет параметры  $min$  и  $max$  и реализует пороговую функцию «прижатия» входного значения к отрезку  $[min, max]$ .

Результат визуализации, соответствующей продемонстрированным на рис. 1 настройкам, показан на рис. 2. За основу взят решатель Burgers2 задачи об оптимизации гибридных разностных схем [5].

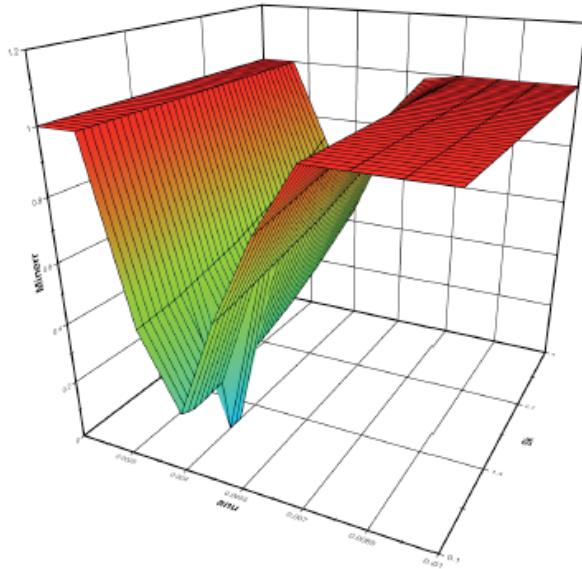


Рис. 2: Результат визуализации сцены, соответствующей настройкам с рис. 1

Онтологический подход к описанию и хранению фильтров позволяет расширять их набор без модификации отлаженного ранее программного кода клиента и сервера, легко адаптируя систему визуализации к новым задачам.

### Тонкий Web-клиент

Нативный клиент SciVi для всех поддерживаемых платформ обладает высокой эффективностью реализации, так как его ядро написано на языке C++, выполняется напрямую на процессоре вычислительного устройства и обладает минимальными накладными расходами на вызовы внутренних функций. Однако, использование нативного клиента не всегда удобно, поскольку требует скачивания и установки на машину клиента.

Чтобы дать пользователю возможность гибкой и удобной работы с системой SciVi, было принято решение в дополнение к нативному клиенту реализовать также тонкий браузерный клиент, в котором средства визуализации основаны на стандарте WebGL. На сегодняшний день данный стандарт поддерживается почти всеми современными браузерами как для настольных компьютеров, так и для мобильных устройств. Таким образом, тонкий клиент не нарушает мультиплатформенность.

В качестве графического движка, на основе которого работает тонкий клиент, было принято решение использовать Tree.js [4] ввиду хорошей документированности и высокой стабильности функционирования приложений на базе этой библиотеки. С использованием Tree.js реализован новый модуль визуализации графических сцен SciViRenderer, функциональность которого аналогична соответствующему модулю визуализации в составе нативного клиента.

Графический интерфейс пользователя (GUI) тонкого клиента создан с использованием HTML5 и CSS.

Стек сторонних и разработанных авторами программных средств, на базе которых работает тонкий клиент, представлен на рис. 3.

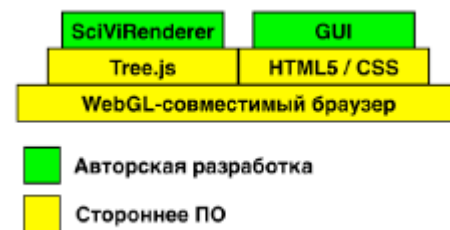


Рис. 3: Программные средства, лежащие в основе тонкого Web-клиента системы научной визуализации SciVi

Наличие тонкого Web-клиента позволяет пользователю без специальной подготовки быстро воспользоваться функциональностью системы SciVi с любого компьютера или мобильного устройства, даже без прав на администрирование, без установки дополнительного программного обеспечения.

При наличии постоянно функционирующего сервера, система SciVi может работать как полноценное SaaS (англ. *Software as a Service*) приложение.

Следует, однако, отметить, что тонкий клиент обладает несколько меньшей производительностью по сравнению с нативным, а также, в отличие от нативного, не может работать в автономном режиме (без подключения к сети Интернет). Более низкая производительность объясняется тем, что алгоритмы рендеринга реализованы на интерпретируемом языке JavaScript и исполняются в среде браузера. Таким образом пользователь имеет возможность выбора типа клиентской части приложения в зависимости от специфики решаемых задач и инфраструктуры программно-аппаратного обеспечения.

### Заключение

В данной статье описаны основные направления развития адаптивной мультиплатформенной

клиент-серверной системы научной визуализации SciVi, построенной на принципах онтологического инжиниринга: разработка подсистемы семантических фильтров и реализация тонкого клиента. Семантические фильтры позволяют гибко настраивать SciVi на специфику решаемой задачи, освобождая пользователя от необходимости изменять исходный код решателя в том случае, если требуется оперативно по некоторому алгоритму преобразовать генерируемые данные перед визуализацией (например, применить к ним некоторую оконную функцию).

Тонкий (браузерный) клиент позволяет использовать SciVi без установки на компьютер пользователя какого-либо специализированного программного обеспечения; требуется лишь наличие WebGL-совместимого браузера и подключения к сети Интернет.

Дальнейшее развитие системы SciVi предполагает расширение базы поддерживаемых способов визуализации и семантических фильтров с целью сделать эту систему пригодной для решения как можно более широкого круга реальных научных задач из различных областей знания.

## Литература

- [1] Lee B., Hudson A.R. Issues in Dataflow Computing // *Advances in Computers*. – Elsevier, 1993. – Vol. 37. – pp.285–333
- [2] Ryabinin K., Chuprina S. Development of Ontology-Based Multiplatform Adaptive Scientific Visualization System // *Journal of Computational Science*. – Elsevier, 2015. – Vol. 10. – pp.370–381
- [3] Ryabinin K., Chuprina S. Using Scientific Visualization Tools to Bridge the Talent Gap // *Procedia Computer Science*. – Elsevier, 2015. – Vol. 51. – pp.1734–1741
- [4] Библиотека Tree.js [Электронный ресурс]. URL: <http://threejs.org/> (дата обращения: 05.05.2016)
- [5] Бондарев А.Е., Бондаренко А.В., Галактионов В.А., Михайлова Т.Н., Рыжова И.Г. Разработка инструментального программного средства Burgers2 для оптимизации гибридных разностных схем // *Научная визуализация*. – М.: Национальный исследовательский ядерный университет МИФИ, 2013. – К. 1, Т. 5, №1. – С.26–37
- [6] Рябинин К.В., Чуприна С.И. Адаптация систем научной визуализации к сторонним решателям // *Труды Юбилейной 25-й Международной конференции GraphiCon2015*. – М., 2015. – С.127–131

## Об авторах

Константин Валентинович Рябинин, к.ф.-м.н., преподаватель кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета.

Светлана Игоревна Чуприна, к.ф.-м.н., заведующая кафедрой математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета.

Антон Юрьевич Бортников, студент кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета.