

# R&D ЛАБОРАТОРИЯ “ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В НАУЧНЫХ ИССЛЕДОВАНИЯХ”

<http://itis.cs.msu.ru>

Мы занимаемся разработкой программного обеспечения для проведения научных исследований, в том числе вычислительного моделирования, интерактивной визуализации и анализа информации.

С 2008 года мы ведем активное сотрудничество с технологическими группами Microsoft Research Cambridge, с 2010 года-работаем над проектами РКК Энергия, с 2011 года сотрудничаем с Microsoft Research Redmond. Все основные участники команды прошли стажировку в Microsoft Research Cambridge.

Руководитель лаборатории – Сергей Березин ([sergey@itis.cs.msu.ru](mailto:sergey@itis.cs.msu.ru)), к.ф.-м.н., доцент МГУ.

## ТЕХНОЛОГИИ

- Языки программирования: F#, C#, C++, JavaScript, TypeScript, Python, Java
- Платформы: Microsoft .NET, Mono, Linux, HTML5, Microsoft Azure
- Web технологии: ASP.NET, SignalR, OWIN, LESS
- Асинхронное, параллельное и конкурентное программирование: TPL, F# Async, CCR, MPI, AMP, OpenMP, CUDA
- Графические библиотеки: OpenGL, DirectX, WebGL
- Пользовательский интерфейс: WPF, HTML5, Qt
- Базы данных: SQL, NoSQL, RDF

## ПРОЦЕСС РАЗРАБОТКИ

- Методологии SCRUM и Канбан
- Программные средства: VisualStudio.com, Git/GitHub, Taiga, Trello
- Опыт работы в распределенном коллективе: МГУ + Redmond + UC Berkeley + Казань

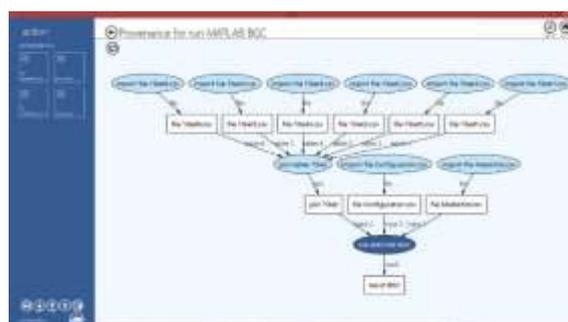
## DISTRIBUTION MODELLER

<http://research.microsoft.com/en-us/projects/msrceesdm>

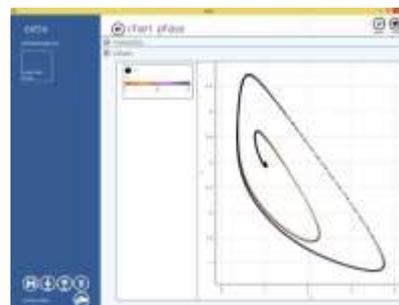
<http://research.microsoft.com/en-US/projects/msrceesdm/altai.aspx>

Distribution Modeller (DM) – платформа для выполнения вычислительных экспериментов, позволяющая исследователям загружать исходные данные из файлов или облачных сервисов, таких как FetchClimate, создавать многопараметрические модели при помощи графического конструктора или на языке F#, выполнять байесовскую оценку параметров модели, строить визуализации полученных результатов и сохранять полное описание эксперимента и данных для последующего распространения, воспроизведения и модификации.

Отличительными чертами DM являются поддержка истории происхождения данных (provenance), обеспечение воспроизводимости эксперимента и поддержка работы с неопределенностью (вероятностными данными) на всех этапах вычислительного эксперимента.



Вычислительное ядро DM написано на C# и F#. Большая часть компонентов пользовательского интерфейса написана на JavaScript и TypeScript, что обеспечивает одинаковый интерфейс приложения Windows и HTML страниц, содержащих компоненты DM, а также позволяет строить на основе DM распределенные системы, в которых вычислительная и интерфейсная части работают на разных компьютерах. Для взаимодействия между компонентами .NET и JavaScript применяется технология SignalR.



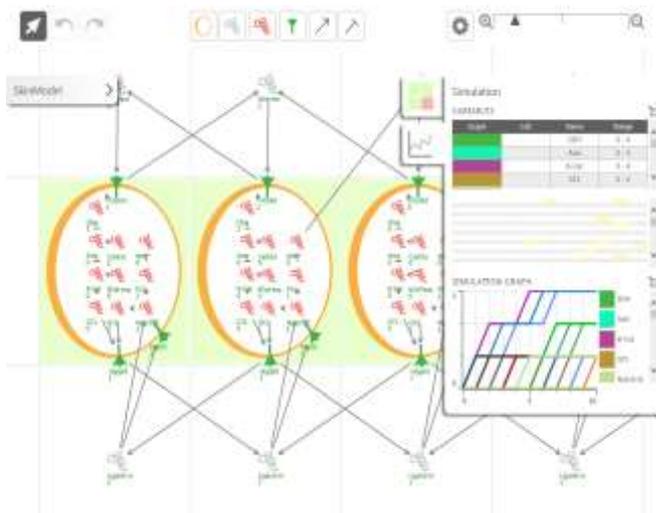
## BIO MODEL ANALYZER

<http://biomodelanalyzer.research.microsoft.com>

Облачный сервис с интуитивным интерфейсом, позволяющий создавать схемы сигнальных взаимодействий в биологических моделях и проводить анализ поведения и устойчивости моделей в различных условиях.

Графический пользовательский интерфейс работает на всех устройствах, поддерживающих HTML5 и предоставляет средства удобного построения сложных биологических моделей с применением touch-интерфейса и drag'n'drop.

Анализ модели выполняет код F#, работающий на автоматически масштабируемых серверах в облаке Microsoft Azure.



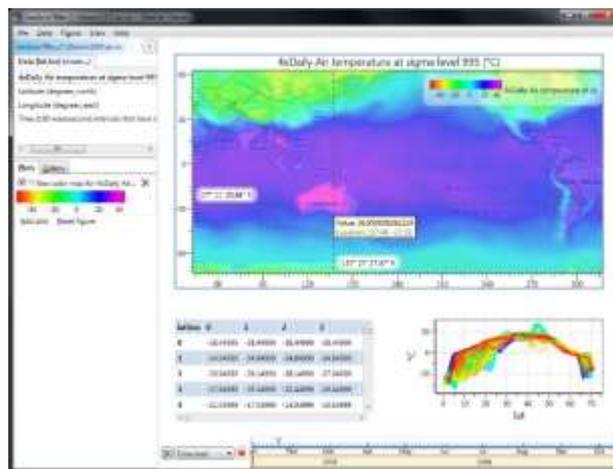
## SCIENTIFIC DATASET & DATASET VIEWER

<http://research.microsoft.com/en-us/projects/sds>

<http://research.microsoft.com/en-us/um/cambridge/groups/science/tools/datasetviewer/datasetviewer.htm>

Scientific DataSet (SDS) – библиотека классов на платформе .NET для работы с многомерными наборами данных, такими как временные ряды, матрицы, климатические данные, спутниковые и медицинские изображения, результаты гидродинамического моделирования и многие другие.

Библиотека SDS поддерживает несколько распространенных форматов файлов: текстовые файлы с разделителем (CSV), двоичные кросс-платформенные форматы NetCDF и HDF5. SDS позволяет задавать описания наборов данных при помощи метаданных, поддерживает параллельный доступ к наборам данных и масштабирование от отдельного текстового файла до хранилища в облаке Windows Azure.



DataSet Viewer - программное обеспечение для удобной визуализации многомерных данных, написанное на C# и использующее библиотеки WPF и DirectX. DataSet Viewer использует возможности DirectX и графических процессоров для отображения наборов данных, содержащих миллионы точек. Возможно DataSet Viewer включают возможности выбора подмножества и среза многомерного массива, построения различных типов графиков, в том числе над географическими картами. Построенные визуализации могут быть экспортированы как растровые изображения, видеофрагменты или файлы специального формата, содержащие данные и визуализации одновременно.

## MICROSOFT RESEARCH FETCHCLIMATE:

<http://fetchclimate2.cloudapp.net>

Microsoft Research FetchClimate – сервис на платформе Microsoft Azure для предоставления климатических данных исследователям в различных прикладных областях. Включает в себя интерактивный графический клиент для получения и отображения климатической информации, построенный на технологиях HTML5, JavaScript, LESS и вычислительный сервис для хранения и обработки данных, написанный на C# и C++. Возможен программный доступ к сервису по протоколу HTTP. Предусмотрены клиентские библиотеки для платформы .NET, JavaScript и статистического пакета R.

Исходными данными являются доступные в Интернет базы данных с климатической информацией. В целях увеличения эффективности, многомерные массивы исходных климатических данных были разбиты на блоки небольшого размера, которые скопированы в хранилище Azure Blobs сервиса FetchClimate.

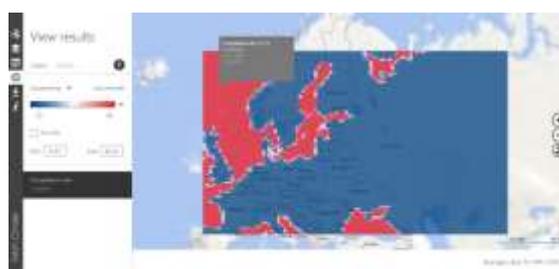
При выполнении запроса FetchClimate оценивает величину потенциальной ошибки (неопределенность) для всех имеющихся источников климатической информации и объединяет информацию из нескольких источников для уменьшения неопределенности ответа. Вычислительный сервис обеспечивает распараллеливание больших запросов для обеспечения наилучшей загрузки рабочих ролей.



*Выбор климатического параметра*



*Среднегодовое количество осадков*



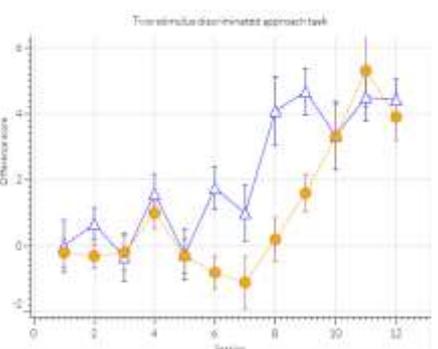
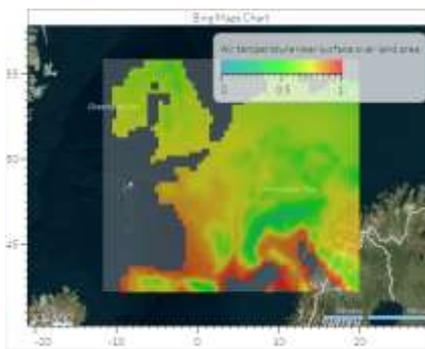
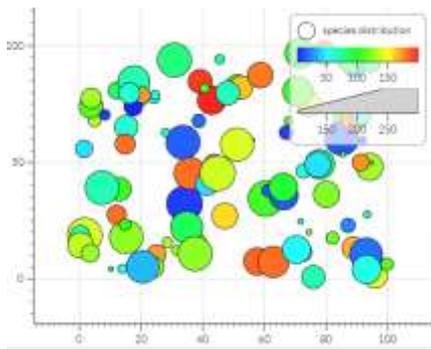
*Оценка неопределенности данных*

## INTERACTIVE DATA DISPLAY / DYNAMIC DATA DISPLAY

<https://github.com/predictionmachines/InteractiveDataDisplay>

<http://dynamicdatadisplay.codeplex.com>

Библиотека компонент для интерактивной визуализации научных данных. Особенное внимание уделяется точности отображаемой информации и производительности при отображении массивов данных большого размера. Основной идеей Interactive Data Display является отображение данных на бесконечном двумерном “холсте” с возможностью приближения, удаления, отображения осей. Графики включают различные варианты маркеров, цветовые карты, географические карты, визуализацию данных с неопределенностью (вероятностных данных). Поддерживается touch-интерфейс и многопоточный рендеринг для обеспечения интерактивной плавной анимации.



Открытый исходный код InteractiveDataDisplay для платформы JavaScript опубликован на GitHub. Предыдущие версии DynamicDataDisplay для платформ Silverlight и WPF были скачаны более 100 000 раз. Компоненты InteractiveDataDisplay применяются во многих разработках лаборатории, в том числе ChronoZoom, FetchClimate, BioModelAnalyzer и Distribution Modeller.

## BIOGEOCHEMISTRY DATA SYSTEM

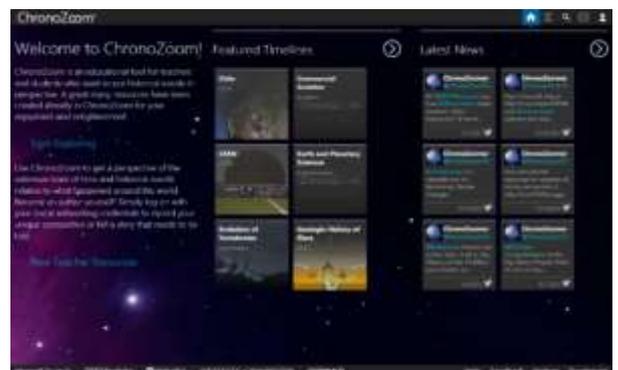
BioGeoChemistry Data System (BDS) – размещенная в облаке Microsoft Azure информационная система для сбора, обработки и поиска данных по растворенным в воде органическим соединениям. Сервис BDS ориентирован на исследователей, работающих в области биогеохимии - междисциплинарной науки о геохимических процессах, происходящих в биосфере Земли при участии живых организмов. BDS позволяет исследователю загружать в систему свои данные, снабжать данные описанием в форме метаданных, применять несколько методов обработки данных, типичных для биогеохимии, и выполнять поиск, используя в качестве критерия пространственно-временные координаты образцов, химическую формулу соединения или метаданные.

BDS использует компоненты Distribution Modeller и обеспечивает сохранение полной информации о происхождении данных и воспроизводимость любого результата, присутствующего в системе. Код серверной части написан на F#, пользовательский web-интерфейс использует TypeScript и библиотеку KnockoutJS. Часть вычислительных алгоритмов написана на MATLAB и работает на выделенной виртуальной машины. Во время бездействия системы виртуальная машина приостанавливается, с целью минимизации затрат на эксплуатацию системы.

## CHRONOZOOM

<http://www.chronozoom.com>

ChronoZoom – веб-приложение с интерактивной визуализацией истории Вселенной, Земли и человечества, от большого взрыва до наших дней для широкого круга пользователей, от школьников до историков. Все исторические события размещены на едином “холсте”. Пользователь может перемещаться вдоль временной оси, изменять масштаб, приближаться к отдельным событиям или рассматривать целые эпохи. Анимация применяется для сохранения контекста пользователя при путешествии во времени. С каждым событием может быть связано произвольное количество мультимедийной информации: изображений, видеофрагментов, документов, которые отображаются при приближении к событию. Возможно совмещение на одном экране информации о событиях и графиков, например, среднегодовой температуры. Поддерживаются экскурсионные туры с аудио сопровождением.



ChronoZoom позволяет зарегистрированным пользователям создавать свои коллекции событий. Применение технологии Azure Access Control System позволяет использовать для регистрации учетные записи Microsoft, Google и др. Интеграция с сервисами Microsoft Bing и Microsoft OneDrive позволяет быстро находить в Интернет и добавлять информацию об исторических событиях. Веб-интерфейс ChronoZoom построен на технологиях HTML5, TypeScript, LESS и размещен в облаке Microsoft Azure. База данных исторических событий работает на SQL Azure, растровые изображения хранятся в блоках.

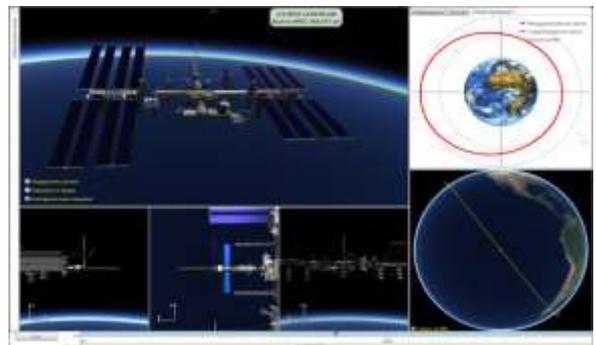


## SPACE ENGINE

Трёхмерный графический движок для визуализации интерактивных трёхмерных моделей. Ориентирован на визуализацию космических полетов и поддерживает одновременную визуализацию моделей разного масштаба, начиная от отдельных зданий и космических аппаратов до земного шара и орбит космических аппаратов.



Space Engine написан на языке C# с применением DirectX. Трёхмерная модель Земли использует иерархические тайловые карты для реалистичного отображения поверхности земного шара на разных уровнях детализации.



На основе Space Engine для корпорации Энергия были написаны несколько интерактивных приложений, отображающих вывод на орбиту и маневры космических аппаратов на основе поступающей в реальном времени телеметрической информации.

## MULTIVIEWER

MultiViewer – программное обеспечение для интерактивной 2D и 3D визуализации на видеостенах – матрицах из большого количества мониторов, созданное в рамках совместного проекта с Институтом космических исследований



Особенностью MultiViewer является поддержка кластера из нескольких серверов, каждый из которых обслуживает два или четыре монитора. Распараллеливание задач загрузки данных и рендеринга позволяет обеспечить производительность, достаточную для интерактивного построения визуализаций сверхвысокого разрешения. Программное обеспечение написано на C#, для работы с графикой применяются библиотеки WPF и DirectX, синхронизация кластера осуществляется средствами Windows Communication Foundation.