

1 Введение

Тут должен быть краткий обзор предметной области (о моделировании (краткая классификация), мы выбираем имитационное моделирование, мелкозернистый параллелизм, его использования для моделирования) Цель, актуальность, план следующего изложения.

Необходимость постановки опыта (или эксперимента) над системой (под системой понимается совокупность объектов, функционирующих и взаимодействующих друг с другом для достижения определённой цели – в [2] написано, что такое определение даётся в [Шмидт, Тэйлор, 1970]) возникает постоянно и в самых различных сферах деятельности, будь то физические процессы, экономические системы и т.д. Однако, во-первых, произвести эксперимент над реальной системой не всегда представляется возможным в силу того, что системы может либо не существовать, либо в результате эксперимента будет невозможным её возвращение в исходное состояние, а, во-вторых, эксперимент над реально существующей системой может потребовать слишком больших затрат на его осуществление. В таких случаях возможно прибегнуть к моделированию: построить модель – упрощённый вариант системы (что-нить о том, что этот вариант должен обладать требуемым свойствами) – и провести эксперимент уже над моделью. На этапе построения модели необходимо сделать выбор между физической моделью (создание упрощённого, но реального существующего варианта системы для проведение эксперимента) и математической (описание системы посредством отношений, которые определяют как система будет реагировать на изменения, если бы она существовала). Математическая модель может иметь точное аналитическое решение, дающее представление о том, как входные параметры влияют на систему в соответствии с построенной моделью. Однако описываемые модель отношения могут не дать простого аналитического решения и потребовать огромных вычислительных ресурсов, что сделает невозможным аналитическое моделирование или потребует большего упрощения системы. Альтернативный вариант в таком случае: изучение системы с помощью имитационного моделирования, то есть многократного испытания модели с нужными входными данными. Имитационные модели за последнее время получили широкое распространение благодаря появлению инструментов, упрощающих создание компьютерных программ, и увеличению и удешевлению компьютерных мощностей. [2, стр. 23-34] сказать, что моделирование может быть физическим, аналитическим, но мы будем рассматривать имитационное. Однако, моделирование невозможно без инструмента визуализации, позволяющего изучать модель и исследовать результаты смоделированного эксперимента: работа с объектами на данных микроуровне чрезвычайно сложна, т. к. он представляет из себя огромный массив ячеек с частицами, в то время как визуализация позволяет получить качественную картину ????. Целью данной работы является разработка подсистемы пользовательского интерфейса и визуализации данных среды имитационного моделирования, назначение которой – построение и исследование клеточно-автоматных моделей, широко применяющихся для исследования явлений

естественных наук. К системе предъявляются требования,

Цель работы — обеспечить исследователя МЗП моделей физических процессов инструментом для построения и отладки таких моделей и визуализации данных в них. Хотя система должна обладать достаточной универсальностью и быть пригодной для широкого спектра таких моделей, в первую очередь рассматривается визуализация данных для клеточно-автоматных моделей газов. Актуальность. Рассматриваемый класс моделей характерен сложностью происходящих в них преобразований данных и большим объемом данных. Под большим объемом данных подразумевается, что он существенно превосходит размеры основной памяти. Сложность преобразований данных в моделях означает необходимость: 1) отладки модели в процессе ее создания и 2) создания режимов отображения, позволяющих исследователю увидеть как точную количественную картину некоторого фрагмента состояния модели, так и качественную картину протекающих в модели процессов. Если исследователь выбирает путь построения собственной программы визуализации, он неизбежно сталкивается с необходимостью реализации системных функций, сложность которых значительно выше, чем само описание модели. Таким образом, для эффективной работы исследователя необходим инструмент для построения, отладки и исследования моделей.

Далее в работе будет описан процесс разработки системы по этапам. В первую очередь были предъявлены требования к разрабатываемой системе, главными из которых являются: адекватность предметной области и открытость, в разделе «2 Формирование требований» будет подробно раскрыто каждое из требований, обусловлено наличие их предъявление к создаваемой системе. В разделе «3 Обзор существующих систем» представлен краткий обзор уже имеющихся систем визуализации в исследуемой предметной области, исследование их на соответствие сформулированным ранее требованиям, сравнение по требуемым параметрам. В разделе «4 ???» разобрана архитектура разработанной системы, детально описан каждый компонент системы: тутнуженглагол цель создания данного модуля системы, какие из поставленных задач он решает и каким требованиям удовлетворяет, список основных функций, которые он реализовывает. Надо ли писать про заключение?

2 Формирование требований

Назначение системы — визуализация данных в МЗП моделях физических процессов. Наиболее существенное требование, требование адекватности системы проблемной области, заключается в создании удобного инструмента, адекватного потребностям исследователя широкого спектра таких моделей. Моделирования физических явлений с помощью моделей с мелко-зернистым параллелизмом — динамично развивающаяся область, в которой постоянно возникают новые классы моделей. Для того, чтобы система могла применяться для них, она должна иметь открытую архитектуру, позволяя пополнять свой набор функций не только разработчикам, но и пользователям системы. Сказать про большие объёмы данных?. Таким образом, основными требованиями, предъявляемыми к системе, являются:

- адекватность проблемной области
- открытость
- обработка больших объёмов данных

рассмотрим более подробно, что включает в себя каждое из данных требований.

2.1 Адекватность проблемной области

Адекватность системы предметной области означает удовлетворение системой требованиям, которые возникают из предметной области: так как система создаётся для работы клеточно-автоматными моделями, то она должна реализовывать необходимые функции, требующиеся при исследований данной области, а именно:

- **Работа с моделями.** С точки зрения системы модель представляет из себя непосредственно свойства модели как физического объекта (т. е. Все свойства модели: это могут быть правила, по которым осуществляются переходы в ячейках, массы частиц покоя и количество движущихся частиц, и т.п.) и объект данных, над которым производятся вычисления. Исследователю необходимо иметь возможность формировать и корректировать параметры модели и создавать и редактировать объект данных.
- **Работа с проектами.** В процессе исследования возникает необходимость подкорректировать параметры изучаемой модели, посмотреть на изменение поведения, сравнить данную модели с исходной по каким либо характеристикам. В таких случаях удобно объединять две и более моделей в один проект с возможностью сохранения и загрузки проекта.
- **Исполнение модели.** Одной из основных функций является счёт модели: это включает в себя передачу параметров модели и объекта данных на вычислитель,

запуск модели на фиксированное число итераций, ожидание завершения счёта и последующее получение результирующих данных с вычислителя.

- **Изучение модели:** дать возможность пользователю системы работать с моделью не на микроуровне, а предоставить качественную картину и возможность взаимодействия с системой, т.е. Обеспечить интерактивность. Качественная картина может быть предоставлена за счёт поддержки различных режимов отображения (режимом отображения является функция, отображающая объект данных в изображение, которое впоследствии может быть выведено на экран монитора): например, это может быть режим осреднённых плотностей или профиль волны для исследования её поведения. Полезными могут быть не только режимы, визуализирующие какую-либо конкретную характеристику (плотность частиц, направление потоков), но и режимы, формирующие целостную картину: например, трёхмерный режим с возможностью масштабирования и вращения для просмотра объекта данных с разных ракурсов.
- **Отладка модели.** Во время исследования востребован не только запуск модели на счёт на фиксированное, большое количество итераций, но и отладка модели в режиме пошагового исполнения: формирование модели и отправка её на вычислитель, запуск на небольшое количество шагов, получение результирующих данных с возможностью продолжить вычисления с того момента, где они были приостановлены. Так же востребованной может оказаться возможность модификации текущего объекта данных, передача изменений на вычислитель и продолжение счёта.

2.2 Открытость

Свойство открытости включает в себя:

- **Расширяемость.** Система должна быть готова к изменениям, связанным с включением поддержки новых форматов данных (объяснить что такое формат данных) и режимов отображения, ранее не предусмотренных системой, так как требуется создание универсального инструмента, реализующего не только необходимый на данный момент функционал, но и способного в будущем адаптироваться под новые требования, предоставляющего механизм для встраивания нового функционала, т. е. иметь возможность визуализировать различные классы моделей клеточных автоматов и готового к подключению поддержки новых классов моделей. Помимо поддержки новых форматов данных и новых классов моделей, так же должна быть возможность встраивать поддержку новых графических библиотек (например, отрисовку трёхмерных изображений на ОС Windows делать не с помощью OpenGL, а добавить поддержку Direct3D) и возможность встраивать альтернативные способы коммуникации с вычислителем (пример?). Всё это требует от системы

масштабируемости по архитектуре: возможность вносить изменения, связанные с добавлением нового функционала, при этом минимизируя влияние на другие, уже существующие компоненты системы.

- **Переносимость.** реализация модулей программы не должна быть привязана к какой-либо платформе. В системе будет допускаться привязкам отдельных реализаций некоторых компонентов к определённым платформам, но в целом система должна быть платформонезависимой.

Переносимость бывает разной для разных компонентов системы [3, page 27]: Source code: компоненты должны собираться и запускаться на разных платформах. Data portability: The ability to move stored data from one application platform to another is fundamental to achieving the objective of application portability. User portability ???

- **Интероперабельность.** способность компонентов системы взаимодействовать между собой. (взаимодействие с системой моделирования?) Application Software Interoperability and Application Platform Interoperability
- **Дружественность интерфейса.** (интерфейса пользователя) – графический должен быть простым и интуитивно-понятным. (и несложным?) (программного интерфейса) – (интерфейс должен требовать реализацию (почти) минимального необходимого набора методов)

2.3 Обработка больших объёмов данных

Под возможностью обработки больших объёмов данных понимается способность системы визуализировать объекты данных, размер которых превышает размер оперативной памяти компьютера. Особенность клеточных автоматов состоит в том, что при большем объёме данных...

* Гибкость – слои и сборка из слоёв. * Интерактивность (часть пользовательского интерфейса, часть адекватности) * Модульность – система должна быть разбита на части, каждая из которых представляет собой ?, и реализацию каждой части возможно будет имитировать не внося изменения в остальные. (разные задачи, разнообразие визуализируемых данных, разные средства визуализации)

3 Обзор существующих систем

обзор систем по одной форме (gnuplot, octave, ...). * вывод, что ни одна система не соответствует в полной мере требованиям.

3.1 Mirek's Celebration

Система моделирования 1D и 2D клеточно-автоматных моделей. Имеет удобный, интуитивно понятный пользовательский интерфейс, хорошо документирована. Расширяемость +- Есть возможности встраивания DLL. Возможности расширения всё-таки ограничены. Интероперабельность ??

Платформонезависимость - Win32 Другжественный интерфейс +

Адекватность пред. обл. +

Большие объёмы данных -

3.2 Ansys Fluent

Система моделирования широкого спектра физических процессов, в том числе и газовой динамики, обладает богатыми возможностями моделирования и визуализации.

Расширяемость + Возможность встраивания пользовательских функций. Интероперабельность ??

Платформонезависимость ??

Другжественный интерфейс +

Адекватность пред. обл. - Дифференциальные уравнения решаются классическими методами. Большие объёмы данных ??

4 Описание своей системы / Архитектура

Архитектура системы представляет из себя 4 взаимодействующих друг с другом модулей (или ещё сказать, что есть часть, которая связывает данные модули):

- Взаимодействие с подсистемой моделирования
- Поддержка графических библиотек
- Поддержка различных режимов отображения
- Пользовательский интерфейс

Графически модульная архитектура системы представлена на рисунке номер. Так же рисунок, на котором изображены зависимости между модулями.

Такая модульная/открытая архитектура системы, разбивающая систему на отдельные, взаимодействующие между собой части, реализаций которых одна от другой никак не зависят, позволяет вносить изменения (как модификация существующей реализации, так и расширение за счёт *чего-то нового*) в отдельные компоненты прозрачно для остальных.

4.1 Модуль взаимодействия с подсистемой моделирования

Для обеспечения коммуникации подсистемы визуализации с подсистемой моделирования был создан интерфейс для подключения вычислителей.

В основные функции, решаемые данным модулем, входит запуск процесса моделирования в синхронном (запуск на счёт фиксированного количества итераций, ожидание завершения моделирования и получение результата) и в асинхронном (запуск на счёт с возможностью подключения к вычислителю в любой момент и получения текущего состояния *чего???*) режимах и реализация протокола передачи объекта данных визуализатору с вычислителя.

Класс возможных вычислителей, с которыми может коммуницировать подсистема визуализации, программно ничем не ограничен. Таким образом, в качестве вычислителя может выступать *абсолютно любое устройство*: как процессор или видеокарта локальной машины, так и удалённый кластер, который, возможно, требует подключения по специфичному??? протоколу (вполне стандартна ситуация, когда доступ к кластеру осуществляется по SSH с необходимостью авторизации).

Подключение нового вычислителя осуществляется написанием реализации созданного интерфейса. *Может, ссылку на приложение, которое создать для каждого модуля, где будет кратко описано, как создавать реализацию нового модуля?* На данный момент реализована поддержка разделяемых библиотек для запуска счёта на локальной машине и в процессе разработки поддержка запуска счёта на удалённой машине, в том числе и для моделирования на кластере.

Т.о. система удовлетворяет свойству интероперабельности. <- *надо куда-то красивее вписать.*

4.2 Модуль поддержки графических библиотек

Поскольку основной задачей инструмента визуализации является *отрисовка* изображения, то в состав системы так же был включён модуль, отвечающий за поддержку различных графических библиотек. На данный момент существует множество библиотек, как для работы с двухмерной графикой (GTK, Qt, wxWidgets и многие другие) и с трёхмерной графикой (OpenGL, Direct3D и т.д.) и основной задачей модуля является предоставление общего интерфейса для рисования независимо от того, какая из библиотек используется в данный момент для формирования итогового изображения.

Модуль поддержки графических библиотек, как и модуль взаимодействия с подсистемой моделирования, представляет из себя интерфейс, реализацией которого осуществляет подключения новых библиотек. В интерфейс входит рисование графических 2D и 3D примитивов (точки, линии, закрашивание областей), рисование более сложных объектов (различного рода кривые, объединение и пересечение объектов), поддержка прозрачности, если таковая есть в библиотеки. Иными словами, задачей интерфейса является предоставление *универсального* доступа к возможностям библиотек, скрыв при этом различия при работе с ними для других модулей. *Сказать про возможную зависимость библиотек и платформ?*

За счёт введения интерфейса достигается расширяемость, а отсутствие привязанности интерфейса к каким-либо конкретным технологиям делает систему переносимой. Стоит отметить, что некоторые графические библиотеки являются платформозависимыми (например, возможность использовать Direct3D имеется только под операционной системой Microsoft Windows), однако, в целом это не делает систему переносимой.

На данный момент используются следующие кроссплатформенные библиотеки:

- Модули QtGUI и QtWidgets из библиотеки Qt.
- Библиотека OpenGL для отрисовки 3D изображений.

Как подключить новые библиотеки?

4.3 Модуль поддержки режимов отображения

В разделе 2.1 одним из требований, предъявляемых к системе, является предоставление исследователю как возможности наблюдать за изменениями отдельных различных характеристик, так и возможности увидеть целостную картину. Данный модуль реализует поддержку различных режимов отображения и предоставляет возможности для встраивания новых режимов.

Единственная функция, которую реализует модуль поддержки режимов отображения, это отображение части или всего объекта данных на двухмерную или трёхмерную картинку. При отрисовывании изображения данный модуль использует интерфейс, предоставляемый модулем поддержки графических библиотек, а работа с объектом данных модели происходит через модуль взаимодействия с подсистемой моделирования.

Расширяемость.

Реализации поддержки режима отображения включает в себя реализацию двух *компонентов*: функцию Написать про возможность выделение слоёв (переиспользуемые компоненты), сборку из слоёв. Для поддержки различных режимов отображения был так же разработан интерфейс для их подключения, что делает систему расширяемой. На данный момент релизованы:

- Режим осреднения (статистически-обобщённый).
- Режим лупы, позволяющий работать с объектами данных на микроуровне.
- 3D режим для получения полноценной картины.

4.4 Модуль пользовательского интерфейса

Для взаимодействия с пользователем системы был реализован графический интерфейс. Задачей модуля было предоставить пользователю простой, но в тоже время функциональный, удобный, интуитивно-понятный интерфейс. Графический интерфейс направлен на удовлетворение свойству адекватности предметной области: через него пользователь осуществляет работу с проектами и моделями,

Представляет из себя MDI интерфейс. Dockable-окна с настройками, всё такое.

5 Заключение

В результате работы была разработана архитектура подсистемы визуализации системы имитационного моделирования, на основе разработанной архитектуры была написана реализация системы. Был разработан и опробован на чём? механизм взаимодействия с моделью. Разработан механизм встраивания режимов визуализации, на базе которого построено несколько режимов каких? Разработана графическая оболочка системы. В планах реализация механизма удалённого исполнения модельных программ, запуск счёта модели на кластере ССКЦ.