



Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЕ ЗАМЕТКИ
по направлению «Разработка систем инженерного анализа и
ресурсоемкого ПО (rndhpc)»

Авторы (исследователи):	Крехтунова Д., Ершов В., Муха В., Тришин И.
Научный(е) руководитель(и):	Соколов А.П., Першин А.Ю.
Консультанты:	@Фамилия И.О.@

Москва, 2021–2021

Работа (документирование) над научным направлением начата 20 сентября 2021 г.

Руководители по направлению:

СОКОЛОВ,	–	канд. физ.-мат. наук, доцент кафедры САПР,
Александр Павлович		МГТУ им. Н.Э. Баумана
ПЕРШИН,	–	PhD, ассистент кафедры САПР,
Антон Юрьевич		МГТУ им. Н.Э. Баумана

Исследователи (студенты кафедры САПР, МГТУ им. Н.Э. Баумана):

Крехтунова Д., Ершов В., Муха В., Тришин И.

C59 **Крехтунова Д., Ершов В., Муха В., Тришин И.. Разработка систем инженерного анализа и ресурсоемкого ПО (rndhpc):** Научно-исследовательские заметки. / Под редакцией Соколова А.П. [Электронный ресурс] — Москва: 2021. — 32 с. URL: <https://arch.rk6.bmstu.ru> (облачный сервис кафедры РК6)

Документ содержит краткие материалы, формируемые обучающимися и исследователями в процессе их работ по одному научному направлению.

Документ разработан для оценки результативности проведения научных исследований по направлению «Разработка систем инженерного анализа и ресурсоемкого ПО» в рамках реализации курсовых работ, курсовых проектов, выпускных квалификационных работ бакалавров и магистров, а также диссертационных исследований аспирантов кафедры «Системы автоматизированного проектирования» (РК6) МГТУ им. Н.Э. Баумана.

RNDHPC



Крехтунова Д., Ершов В., Муха В., Тришин И.,
Соколов А.П., Першин А.Ю., 2021

Содержание

1	Теоретические основы графоориентированного программного каркаса	4
2022.01.01:	Отличие сетевых моделей от сетевых графиков	4
2	Разработка графоориентированного дебаггера	4
2021.11.10:	Автоматизированные и автоматические методы отладки, применяемые при реализации сложных вычислительных методов (CBM), отладка наукоёмкого кода, science code debugging, graph based programming и пр. (первичный обзор литературы)	4
	Анализ взвешенных графов вызовов для локализации ошибок программного обеспечения	4
	Интегрированная отладка моделей Modelica	6
	Систематические методы отладки для масштабных вычислительных фреймворков высокопроизводительных вычислений	7
	Ориентированный на данные фреймворк для отладки параллельных приложений	8
	Определение степени и источников недетерминизма в приложениях MPI с помощью ядер графов	10
2021.11.22:	Web-инструменты для древовидного представления ассоциативных массивов, визуализация ассоциативных массивов (первичный обзор литературы)	12
	Ассоциативные массивы	12
	Набор библиотек TnT для web-визуализации деревьев и аннотаций на основе трассировок	13
	Программное обеспечение Empress интерактивного и исследовательского анализа многомерных наборов данных на основе деревьев	14
	Программный инструмент Treemap визуализации иерархических структур	14
	Web-инструмент DoubleRecViz для визуализации согласования деревьев транскриптов и генов	15
2021.12.19:	Концептуальная постановка задачи	16
3	Разработка web-ориентированного редактора графовых моделей	17
2021.10.05:	Обзор языка описания графов DOT	17
2021.12.04:	Краткое описание алгоритма визуализации графа	17
2021.12.05:	Описание текущего состояния проекта	22
4	Реализация поддержки различных стратегий распараллеливания в рамках графоориентированного программного каркаса	26
2021.11.14:	Известные алгоритмы поиска циклов в ориентированных графах	26
5	Графоориентированная методология разработки средств взаимодействия пользователя в системах автоматизированного проектирования и инженерного анализа	29

2021.11.06: Особенности применения графового описания процессов обработки данных в pSeven (DATADVANCE)	29
Принципы функционирования	29
Поддержка циклов и ветвлений	30
Особенности выполнения расчётных схем	30

1 Разработка графоориентированного дебаггера

2021.09.19: Содержание научно-исследовательской заметки

Заметка размещается в рамках L^AT_EX-подраздела (`\subsection`).

В состав заметки следует включать:

- заметку следует создавать с помощью вспомогательной команды `\notestatement{@prjsid@}{@NoteTitle@}`;
- атрибуты заметки (дата, автор, идентификатор исследовательского проекта, тема заметки) следует заполнять явно, без введения дополнительных макроподстановок;
- **рекомендуется** в состав заметки включать: рисунки; схемы; графические результаты расчетов; формулы; математические постановки задач, представляемые исключительно в математически строгом виде;
- при включении в состав заметки утверждения следует добавлять сноску с выходными данными источника (при этом следует добавлять соответствующий источник в файл библиографии `bibliography.bib`);
- все сопроводительные документы по текущей заметке следует размещать в каталоге, имеющем такое же имя, как имя файла заметки (рис. ??);
- объём одной заметки: не более 2-3 страницы.

Подготовлено: *Соколов А.П. (РК6), 2021.09.19*

2 Разработка web-ориентированного редактора графовых моделей

2021.10.05: Обзор языка описания графов DOT

Язык описания графов DOT предоставляется пакетом утилит Graphviz (Graph Visualization Software). Пакет состоит из набора утилит командной строки и программ с графическим интерфейсом, способных обрабатывать файлы на языке DOT, а также из виджетов и библиотек, облегчающих создание графов и программ для построения графов. Более подробно будет рассмотрена утилита `dot`.

`dot` - инструмент для создания многоуровневого графа с возможностью вывода изображения полученного графа в различных форматах (PNG, PDF, PostScript, SVG и др.).

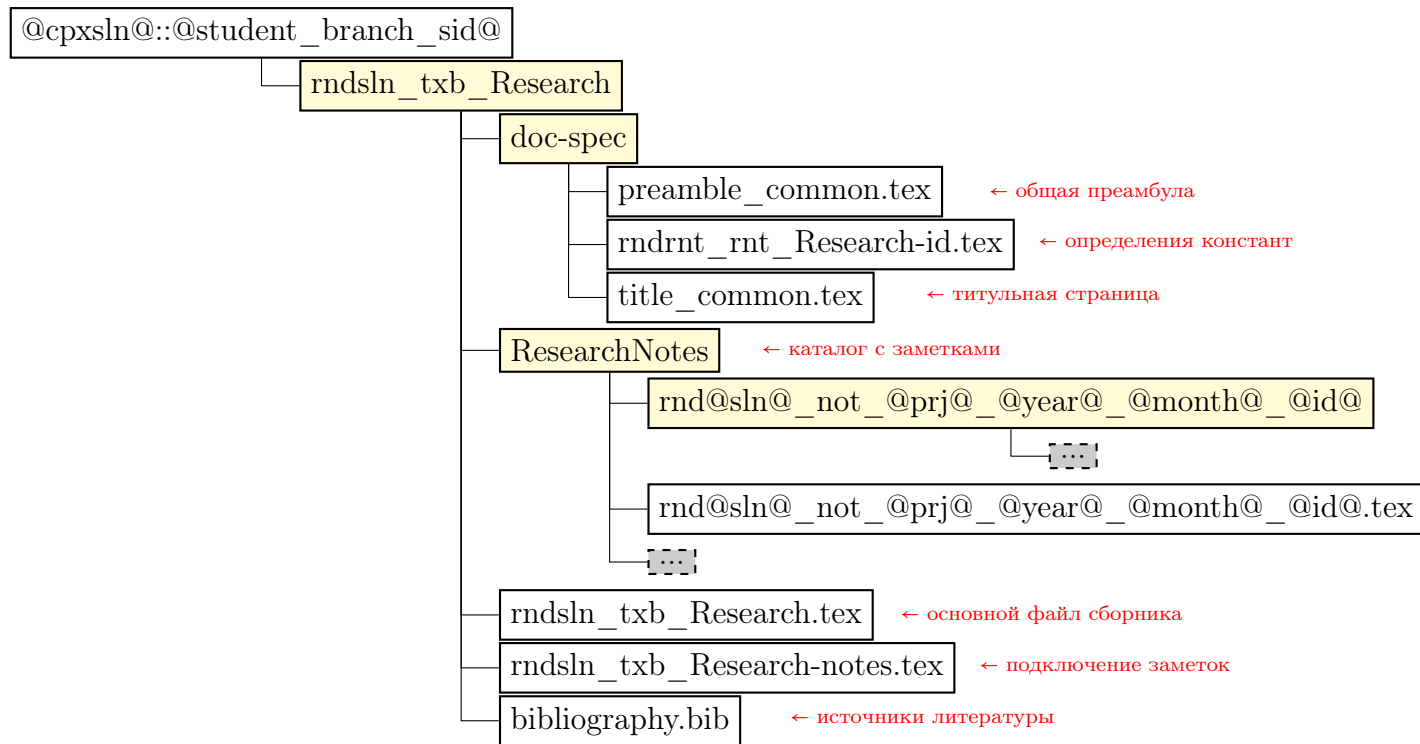


Рис. 1. Структура файловой системы исходников сборника исследовательских заметок

Установка graphviz

Linux: `sudo apt install graphviz`

MacOS: `brew install graphviz`

Вызов всех программ Graphviz осуществляется через командную строку, в процессе ознакомления с языком использовалась следующая команда

`dot -Tpng <pathToDotFile> -o <imageName>`

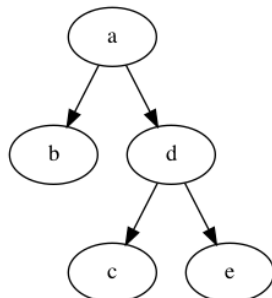
В результате выполнения этой команды будет создано изображение графа в формате png

Пример описания простого графа

```

digraph G {
  a -> b;
  a -> d -> c;
  d -> e;
}

```



Более подробная информация с примерами представлена в обзоре литературы, который находится по следующему пути:

01 - Курсовые проекты/2021-2022 - Разработка web-ориентированного редактора графовых моделей /0 - Обзор литературы/

Подготовлено: *Ершов В. (РК6-72Б), 2021.10.05*

3 Графоориентированная методология разработки средств взаимодействия пользователя в системах автоматизированного проектирования и инженерного анализа

2021.11.06: Особенности применения графового описания процессов обработки данных в pSeven (DATADVANCE)

pSeven – это платформа для анализа данных, оптимизации и создания аппроксимационных моделей, дополняющая средства проектирования и инженерного анализа. pSeven позволяет интегрировать в единой программной среде различные инженерные приложения, алгоритмы многодисциплинарной оптимизации и инструменты анализа данных для упрощения принятия конструкторских решений[19].

Принципы функционирования

На концептуальном уровне в pSeven вводятся следующие понятия:

- Проект – набор файлов, используемых в pSeven для описания решений одной или нескольких задач и хранения результатов их решения.
- Расчётная схема (workflow) – формальное описание процесса решения некоторой задачи в виде ориентированного графа, узлами которого являются блоки, а рёбрами - связи. Такое описание хранится в бинарном файле с расширением .p7wf, использующем некоторый специализированный формат хранения подобного рода описаний.
- Блок – функциональный элемент расчётной схемы, отвечающий за обработку входных данных и формирование выходных данных.[20]
- Порт – переменная определённого типа, описанная в блоке и имеющая в нём уникальное имя, значение которой может быть передано в другие блоки или получено от них через связи.

- Связь (link) – одностороннее соединение между двумя портами, обеспечивающее передачу данных от одного к другому.

Проекты в pSeven имеют единую базу данных, куда записываются все результаты запусков расчётных схем и откуда берутся данные для их последующей презентации пользователю и их анализа. Для определения переменных, значения которых должны быть записаны в неё записаны, предусмотрены специализированные порты для самих расчётных схем, с которыми связываются те блоки, результаты выполнения которых интересуют пользователя.

Связи служат для маршрутизации данных. С их помощью осуществляется и взаимодействие между блоками и, кроме того, определяется очерёдность их запуска. В момент добавления связи в пакете pSeven выполняется проверка портов на совместимость. Они считаются совместимыми, если тип данных источника можно преобразовать к типу данных адресата.[20]

Поддержка циклов и ветвлений

Расчетная схема может включать расчетные циклы. Для их создания применяются специализированные блоки, имеющие функциональную возможность управления запуском других блоков в теле цикла и принятия решения о его прекращении. Такие блоки называются управляющими блоками циклов. Одним из характерных примеров является оптимизационный цикл, управление которым осуществляется из блока Optimizer [20], позволяющего, например, настроить максимальное число итераций или требуемую точность, как условия окончания.

Если речь идёт о задачах анализа данных, существует отдельный блок, обозначающий входную точку цикла (Loop). Такой цикл принимает на вход список наборов аргументов, каждый из которых будет обработан на соответствующем шаге цикла, и выдаёт список наборов выходных значений, которые сохраняются в базе данных проекта.

Кроме того, существует возможность включения в расчётную схему условного и безусловного ветвления. Первое достигается за счёт создания связей для подключения одного и того же порта вывода к различным портам ввода. В этом случае по каждой связи передается копия выходных данных, полученных у источника [20]. Условные ветвления создаются с помощью специального блока Condition, который по определённому условию передаёт входные данные одному из подключенных блоков. Их целесообразно использовать для устранения ошибок в работе блока, отбраковки некорректных входных данных и других аналогичных целей[20].

Особенности выполнения расчётных схем

При выполнении расчётных схем каждый блок запускается в отдельном процессе на уровне операционной системы. Как сказано ранее, начало выполнения блока определяется его связями с другими. Любой блок будет ожидать завершения работы другого блока только в том случае, если ему необходимо получить от него входные данные. Это

означает, что два блока, не имеющих связей друг с другом Блоки, входящие в состав различных ветвлений расчетной схемы, могут запускаться параллельно, поскольку они не зависят друг от друга по используемым данным[20].

Кроме того, при обработке больших выборок данных может потребоваться обрабатывать по несколько наборов данных одновременно в независимых потоках исполнения. Помимо прочего этой цели служит блок **Composite**, который является контейнером для нескольких блоков. В его настройках можно включить опцию параллельного исполнения, указать, с какого порта данные будут обрабатываться параллельно, и указать максимальное число потоков. При запуске расчетной схемы пакет **pSeven** создает несколько виртуальных экземпляров блока **Composite** и автоматически распределяет входные наборы данных между ними. Как только в одном из таких виртуальных блоков завершается расчет, он получает из выборки следующий набор для обработки.[21]

Подготовлено: Тришин И.В. (ПК6), 2021.11.06

2022.02.09: Сравнительная характеристика GBSE, pSeven, Pradis

Для проведения сравнения с GBSE [13] были выбраны следующие известные программные комплексы:


1. Pradis – разработка отечественной компании “Ладуга”;
2. pSeven – разработка отечественной компании “DATADVANCE”.

Выделение признаков для сравнения

При выделении сравнительных признаков необходимо было, чтобы они охватывали достаточно широкую область сведений о программном продукте. Среди прочих должны были быть выделены признаки, относящиеся как к общей структуре программного комплекса, так и к особенностям реализации в нём графоориентированного подхода и, кроме того, к особенностям взаимодействия с пользователем при решении задач, требующих действий со стороны пользователя.

Сравнение осуществлялось с учётом следующих характерных признаков:

1. предметное назначение;
2. принципы формирования графовых моделей;
3. формат описания графовых моделей;
4. особенности работы с входными и выходными данными;
5. особенности передачи данных между узлами графовых моделей;
6. поддержка ветвлений и циклов в топологии графовых моделей;

7. поддержка параллельной обработки данных;
8. возможности отбора результатов расчёта вручную; 
9. возможность доопределять входные данные непосредственно во время обхода графовой модели.

Описание сравниваемых объектов

Pradis

Программный комплекс Pradis, разработанный отечественной компанией «Ладуга», предназначен для анализа динамических процессов в механических системах и системах другой физической природы. Предметом решения являются нелинейные нестационарные задачи. Расчет проводится в функции времени в исходных координатах. Анализ статических задач обеспечивается как частный случай динамического расчета. Круг задач, которые могут быть решены с помощью PRADIS, достаточно широк. Принципиально возможен анализ любых технических объектов, модели поведения которых представимы системой дифференциальных уравнений (ДУ). Практические возможности по анализу конкретных задач определяются текущим составом библиотек комплекса, прежде всего библиотеки моделей элементов.[?]. Данный комплекс был рекомендован к обзору и сравнению, однако после проведенного обзора официальной документации[?], не было получено достаточного представления об использовании графоориентированного подхода в данном комплексе, поэтому было принято решение исключить его из дальнейшего рассмотрения.

pSeven

В программном комплексе pSeven, разработанном компанией DATADVANCE, используется методология диаграмм потоков данных, т.е. топология графа, описывающего процесс решения некоторой задачи проектирования, определяется только зависимостями между входными и выходными данными каждого отдельного процесса их обработки, входящего в решение. [?] В реализованном в pSeven подходе вводятся следующие понятия:

- *Расчётная схема (workflow)* - формальное описание процесса решения некоторой задачи в виде ориентированного графа;
- *Блок* - программный контейнер для некоторого процесса обработки данных, входные и выходные данные для которого задаются через порты;
- *Порт* - переменная определённого типа, имеющая определённое имя, привязанная к блоку;
- *Связь* - направленное соединение типа "один к одному" между входным и выходным портами разных блоков;

С учётом данных понятий можно описать методологию диаграмм потоков данных следующим образом. Расчётная схема содержит в себе набор процессов обработки

данных (блоков), каждый из которых имеет (возможно, пустой) набор именованных входов и выходов (портов). Данные передаются через связи. Для избежания гонок данных множественные связи с одним и тем же входным портом не поддерживаются. Для начала выполнения каждому блоку требуются данные на всех входных портах. Все данные на выходных портах формируются по завершении исполнения блока.[?]

Все порты, которые не привязаны к другим блокам, автоматически становятся внешними входами и выходами для всей расчётной схемы. Для начала обхода расчётной схемы должен быть предоставлен набор входных данных и указаны внешние выходные порты, значения которых обязательно должны быть вычислены в результате обхода. Он производится в несколько этапов: сперва отслеживаются пути от необязательных выходных портов к входным, все встреченные на пути блоки помечаются, как неактуальные и не будут выполнены в дальнейшем; затем отслеживаются пути от обязательных выходных портов к входным и все встреченные на пути блоки помечаются, как обязательные к исполнению. Наконец обязательные к исполнению блоки запускаются, начиная с тех, которые подключены к внешним входам расчётной схемы, а неактуальные игнорируются. Обход прекращается, когда не остаётся необходимых для выполнения блоков. [?]

Сравнительная таблица

Результаты проведённого сравнения были оформлены в общую таблицу, приведённую ниже.

Таблица 1. Сравнительная таблица

Признак	pSeven	GBSE
Спектр задач	Задачи оптимизации, анализ данных	Задачи автоматизированного проектирования, анализ данных
Подход к формированию графа	Согласно описанному в [?] подходу, узлами графа являются блоки, рёбрами - связи, по которым передаются данные.	Узлами графа являются состояния данных, рёбрами - переходы между состояниями, к которым привязываются функции-обработчики. [?]
Формат описания графа	Сформированное описание сохраняется в двоичном файле закрытого формата с расширением .p7wf	Описание графа и функций-обработчиков сохраняется в текстовом файле специального формата .aDOT, являющегося расширением формата DOT[?]
Файловая структура проекта	Проект состоит из непосредственно файла проекта, в котором хранятся ссылки на созданные расчётные схемы и базу данных, сами расчётные схемы, файлы с их входными данными, файлы отчётов, где сохраняются выходные данные последних расчётов и результаты их анализа.	Проект состоит из .aDOT файла с описанием графа, .aINI-файлов с описанием входных данных, библиотеки функций-обработчиков, файлов, куда записываются выходные данные.
Особенности работы с входными и выходными данными	Входные данные должны быть указаны при настройках внешних входных портов расчётной схемы. Данные с выходных портов схемы сохраняются в локальной базе данных. Для их записи в файлы для обработки/анализа вне pSeven необходимо воспользоваться специально предназначенными для этого блоками.	Входные данные хранятся в файле с расширением .aINI, откуда считываются при запуске обхода графа[?]. Для записи выходных/-промежуточных данных в файлы или базы данных необходимо добавить соответствующие функции-обработчики.

Особенности передачи параметров между узлами	Данные между узлами передаются через связи, которые на уровне выполнения создают пространство в памяти для ввода и вывода данных для выполняемых в отдельных процессах блоков. Транзитная передача данных, которые не изменяются в данном блоке, на выход невозможна.	Поскольку узлами графа являются состояния данных, существует возможность задействовать в расчётах только часть данных, оставляя их другую часть без изменений
Поддержка ветвлений и циклов	Присутствует. Достигается за счёт специальных управляющих блоков, которые отслеживают выполнение условий	Присутствует по умолчанию
Поддержка параллельной обработки данных	Присутствует. Блоки, входящие в состав различных ветвлений схемы могут быть выполнены параллельно, поскольку они не зависят друг от друга по используемым данным.	Присутствует. Существует возможность обойти различные ветвления графа одновременно.
Особенности отбора корректных результатов расчёта вручную	Производится на этапе анализа результатов с помощью отчётов, где можно задать фильтрацию выходных данных по указанным параметрам. В случае, если результаты являются промежуточными, расчётную схему приходится разбивать на части.	Планируется реализовать средство визуализации данных, которое вместе с автоматической генерацией форм ввода позволят отбирать корректные результаты промежуточных вычислений во время обхода одного цельного графа.
Возможность доопределения значений входных данных в процессе обхода графа	Отсутствует	Реализована при помощи функций-обработчиков, создающих формы ввода