



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)»

## ОТЧЁТ

о научно-исследовательской работе студента

на тему

«Обзор существующих программных комплексов, реализующих  
графоориентированный подход, и средств взаимодействия пользователя в них»

Студент РК6-71Б  
группа

\_\_\_\_\_  
подпись, дата

Тришин И.В.  
ФИО

Руководитель НИРС

\_\_\_\_\_  
подпись, дата

Соколов А.П.  
ФИО

Консультант

\_\_\_\_\_  
подпись, дата

Першин А.Ю.  
ФИО

Москва, 2021

## РЕФЕРАТ

научно-исследовательская работа студента : 14 с., 8 глав, 0 рис., 0 табл., 9 источн.

Данная работа посвящена обзору литературы в рамках выполнения работы по разработке средств взаимодействия пользователя в автоматизированной системе, реализующей графоориентированный подход к решению исследовательских задач. Был выделен класс задач, в которых необходимо подобное взаимодействие, были рассмотрены некоторые его сценарии. Кроме того в данной работе приводится сравнение рассматриваемой автоматизированной системы с другими присутствующими в данный момент на рынке аналогами в контексте взаимодействия пользователя с процессом решения

**Тип работы:** научно-исследовательская работа студента .

**Тема работы:** *«Обзор существующих программных комплексов, реализующих графоориентированный подход, и средств взаимодействия пользователя в них».*

**Объект исследования:** программные комплексы решения задач проектирования и оптимизации.

## СОКРАЩЕНИЯ

**CASE** Computer-aided Software Engineering. 5

**GUI** Graphical User Interface. 7

**ЛПР** лицо, принимающее решение. 6

## СОДЕРЖАНИЕ

<b>СОКРАЩЕНИЯ</b> .....	3
<b>ВВЕДЕНИЕ</b> .....	5
<b>1. Обзор средств взаимодействия пользователя в графоориентированных системах</b> .....	6
<b>2. Сравнительная характеристика программных комплексов, реализующих графоориентированный подход</b> .....	8
2.1. Выделение признаков для сравнения .....	8
2.2. Описание сравниваемых объектов .....	8
2.3. Сравнительная таблица .....	11
<b>ЗАКЛЮЧЕНИЕ</b> .....	13
<b>Литература</b> .....	14

## ВВЕДЕНИЕ

При проведении современных исследований возникает необходимость автоматизировать процессы решения сложных вычислительных задач. Достижение подобной цели не представляется возможным без формально определенного метода организации процессов в автоматизированной системе. Проектирование, создание и сопровождение подобных систем является трудоемкой задачей, для решения которой применяют инструментальные средства и среды разработки автоматизированных систем (**CASE**-системы)[1] В некоторых очень узко направленных системах подобная организация напрямую зависит от поставленной задачи. В более универсальных же системах, о которых пойдёт речь в этой работе, разрабатывается особая архитектура, которая позволяет организовать различные процессы для решения различных задач по-разному. Подходы к построению данной архитектуры освещены, помимо прочего, в [2].

В данной работе внимание сосредоточено на применении подобных универсальных автоматизированных систем к решению задач, требующих внимания со стороны пользователя непосредственно в процессе решения, и на средствах, которыми взаимодействие с пользователем реализуется.

# 1 Обзор средств взаимодействия пользователя в графоориентированных системах

Среди вычислительных задач, с которыми сталкиваются современные исследователи и разработчики наукоёмкого программного обеспечения, можно выделить те, в которых в результате проведения расчётов получается несколько различных результатов, из которых требуется выбрать наиболее подходящий на основе каких-то критериев или провести ту или иную операцию в зависимости от полученных результатов. В наши дни наблюдается тенденция к автоматизации подобного процесса на основе различных алгоритмов анализа и принятия решений, некоторые из которых разрабатываются специально под конкретную задачу, а некоторые, более универсальные, адаптируются под неё, как, например, описано в [3]. Тем не менее, остаётся широкий спектр задач, где разработка подобных алгоритмов не ведётся ввиду слишком узкой направленности или отсутствии технической возможности автоматизировать принятие решений (как правило, в исследовательских задачах). В таких случаях за него отвечает лицо, принимающее решение (ЛПР). При разработке универсального программного комплекса, позволяющего решать различные задачи проектирования и оптимизации было бы полезно включить возможность ЛПР взаимодействовать с промежуточными результатами вычислений. Помимо прочего, подобная необходимость возникает, когда:

1. нет формально определённых критериев отбора, на основе которых его можно было бы автоматизировать;
2. критериев анализа результатов слишком много для того, чтобы реализовать автоматизированную процедуру для его проведения в пределах исследовательской работы;

Поскольку в данной работе рассматривается, в первую очередь, система, реализующая графоориентированный подход к решению сложных вычислительных задач, то целесообразно рассмотреть подходы к организации взаимодействия пользователя с процессом решения. На основании изложенного выше были выделены следующие сценарии взаимодействия с пользователем в данной системе:

- Введение дополнительных данных, которые требуются на дальнейших этапах расчётов, но которые не были получены автоматически до этого;
- Выбор конкретных данных из некоторого однородного набора для его сужения;
- Выбор дальнейшей логики выполнения расчётов на основании полученных на текущем этапе результатов.

Помимо этого, для эффективной работы с подобной системой исследователю необходим графический пользовательский интерфейс, в котором модель организации вычислений может быть представлена визуально. Большие перспективы в автоматизации процесса решения сложных задач перед исследователем открывает возможность прямого взаимодействия с вычислительной моделью: остановка вычислительного процесса на определенном этапе, изучение обрабаты-

мых данных, просмотр истории изменения обрабатываемых данных, возврат к определенному этапу вычислений, ввод дополнительных параметров на определенной стадии вычислений и т. д.[4] Кроме того, целесообразно рассмотреть основные понятия, вводимые в данной системе.

- *Состояние данных* - некоторый набор данных, в котором они хранятся тройками вида "тип - имя - значение". В GBSE реализован в виде специального класса с названием **Анупар**.
- *Функция-обработчик* - функция, которая вызывается при переходе из одного состояния данных в другое. Фактически данная функция каким-то образом модифицирует объект состояния данных.
- *Функция-предикат* - функция, связанная с тем же переходом, что и некоторая функция-обработчик, проверяющая соответствие входных данных тому формату, в котором они ожидаются на входе обработчика.

На концептуальном уровне абстракции в рассматриваемой системе получение каких-то данных или решений от пользователя может быть реализовано, как и любой другой процесс модификации данных, через соответствующие функции-обработчики и предикаты. Рассмотрим возможные подходы к реализации сценариев взаимодействия пользователя, описанных выше.

Для введения дополнительных данных в GBSE реализован специальный инструмент автоматической генерации графических программных интерфейсов (**GUI**) с формами ввода необходимых данных[5], однако этот инструмент ещё не связан с основным графоориентированным программным каркасом. Средства, реализующие два других сценария на момент написания данной работы находятся в разработке. Для предоставления пользователю возможности сделать выбор относительно дальнейшей обработки данных необходимо разработать следующие средства:

1. Средство визуализации текущего состояния данных
2. Средство содержательной интерпретации текущего состояния данных

Кроме того, для реализации рассматриваемых сценариев будет необходимо доработать средство генерации форм ввода и реализовать в нём дополнительную категорию форм, направленных не на внесение новых данных, а на выбор, в том числе и множественный, из представленных вариантов, отображённых, в том числе, и с помощью средства визуализации.

## 2 Сравнительная характеристика программных комплексов, реализующих графоориентированный подход

Помимо обзора потенциальных сценариев взаимодействия пользователя с системой в процессе обхода графовой модели была проведена сравнительная характеристика рассматриваемой разработки с представленными в настоящее время на рынке продуктами. К сравнению с GBSE были выбраны и рекомендованы следующие программные комплексы:

1. Pradis - разработка отечественной компании "Ладуга"
2. pSeven - разработка отечественной компании DATADVANCE

### 2.1 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 из класса, описанного в разделе 1.

На основании данных требований были выделены следующие признаки для сравнения:

1. спектр задач;
2. подход к формированию графовой модели;
3. формат описания графовой модели;
4. особенности работы с входными и выходными данными;
5. особенности передачи данных между узлами графовой модели;
6. поддержка ветвлений и циклов в топологии графа;
7. поддержка параллельной обработки данных;
8. особенности отбора результатов расчёта вручную;
9. возможность доопределять входные данные непосредственно во время обхода графовой модели.

### 2.2 Описание сравниваемых объектов

#### 2.2.1 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.[6]. Данный комплекс был рекомендован к обзору и сравнению, однако после проведенного обзора официальной документации[7], не было получено достаточного представления об использовании графоориентированного подхода в данном комплексе, поэтому было принято решение исключить его из дальнейшего рассмотрения.

### 2.2.2 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. [8] 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.[8]

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. [8]

### 2.3 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 [8] 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. [9]
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[9]
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[5]. 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 вкупе 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

## ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы были собраны сведения о необходимых для реализации средствах взаимодействия пользователя в системе автоматизированного решения исследовательских задач GBSE, что даёт направление для дальнейшей разработки и программной реализации данных средств. Кроме того, были рассмотрены некоторые существующие на рынке аналоги GBSE, в частности, продукт pSeven, и была проведена сравнительная характеристика данных программных комплексов с учётом возможностей взаимодействия пользователя с процессом решения задач.

## Список использованных источников

- 1 В.О. Голубев. Разработка web-приложений, реализующих бизнес-логику работы пользователя в САПР на основе графоориентированной методологии. 2020.
- 2 Соколов А.П. Першин А.Ю. Система автоматизированного проектирования композиционных материалов. Часть 1. Концепции, архитектура и платформа разработки // Известия СПбГЭТУ «ЛЭТИ». 2020. № 8-9.
- 3 Katal N. Narayan S. Optimal design of QFT controller for pneumatic servo actuator system using multi-objective genetic algorithm // International Journal of Advanced Intelligence Paradigms. 2020. Т. 15, № 2. С. 183–206.
- 4 Соколов А.П. Голубев В.О. Система автоматизированного проектирования композиционных материалов. Часть 3. Графоориентированная методология разработки средств взаимодействия пользователь–система // Известия СПбГЭТУ «ЛЭТИ». 2021. № 2.
- 5 Соколов А.П. Першин А.Ю. Программный инструментарий для создания подсистем ввода данных при разработке систем инженерного анализа // Программная инженерия. 2017. Т. 8, № 12. С. 543–555.
- 6 PRADIS. Общее описание системы [Оф. документация]. 2007.
- 7 PRADIS. Методы формирования и численного расчёта математических моделей переходных процессов [Оф. документация]. 2007.
- 8 Alexey M. Nazarenko Alexander A. Prokhorov. Hierarchical Dataflow Model with Automated File Management for Engineering and Scientific Applications // Procedia Computer Science. 2015. Т. 66. URL: <https://www.sciencedirect.com/science/article/pii/S1877050915034055?pes=vor>.
- 9 Соколов А.П. Першин А.Ю. Графоориентированный программный каркас для реализации сложных вычислительных методов // Программирование. 2018. № X.